

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-158965

(P2002-158965A)

(43)公開日 平成14年5月31日 (2002.5.31)

(51)Int.Cl.⁷

H 04 N 5/91
5/85
5/92

識別記号

F I

H 04 N 5/85
5/91
5/92

テ-マコ-ト^{*} (参考)

B 5 C 0 5 2
N 5 C 0 5 3
H

審査請求 未請求 請求項の数23 O.L (全 74 頁)

(21)出願番号 特願2001-109340 (P2001-109340)
(22)出願日 平成13年4月6日 (2001.4.6)
(31)優先権主張番号 特願2000-183770 (P2000-183770)
(32)優先日 平成12年4月21日 (2000.4.21)
(33)優先権主張国 日本 (JP)
(31)優先権主張番号 特願2000-268043 (P2000-268043)
(32)優先日 平成12年9月5日 (2000.9.5)
(33)優先権主張国 日本 (JP)

(71)出願人 000002185
ソニー株式会社
東京都品川区北品川6丁目7番35号
(72)発明者 加藤 元樹
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内
(72)発明者 浜田 俊也
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内
(74)代理人 100082131
弁理士 稲本 義雄

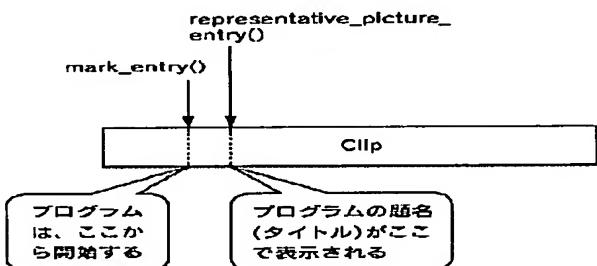
最終頁に続く

(54)【発明の名称】 情報処理装置および方法、記録媒体、プログラム、並びに記録媒体

(57)【要約】

【課題】 A Vストリームの所望の位置に迅速且つ確実にアクセスできるようとする。

【解決ステップ】 A Vストリームを構成するClipのうち、プログラム(番組)の開始点は、mark_entry()に記述され、プログラムのタイトルが表示されているピクチャは、representative_picture_entry()に記述される。



【特許請求の範囲】

【請求項1】 入力されたAVストリームから抽出された特徴的な画像を指示するマークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成するとともに、前記AVストリーム中の所定の区間の組み合わせを定義するPlaylistに対応する再生区間の中から、ユーザが任意に指定した画像を指示するマークから構成されるPlaylistMarkを生成する生成手段と、前記ClipMark、およびPlaylistMarkを各々独立したテーブルとして記録媒体に記録する記録手段とを有することを特徴とする情報処理装置。

【請求項2】 前記生成手段は、前記ClipMarkをClipMarkInformationファイルとして生成するとともに、前記PlaylistをPlaylistファイルとして生成することを特徴とする請求項1に記載の情報処理装置。

【請求項3】 前記PlaylistMarkは、前記Playlistを再生するときのResume点を示すマークをさらに含むことを特徴とする請求項1に記載の情報処理装置。

【請求項4】 前記Playlistを再生するとき、前記Playlistの再生区間に応する前記AVストリームのClipMarkを構成する前記マークを参照することを特徴とする請求項1に記載の情報処理装置。

【請求項5】 前記PlaylistMarkの前記マークは、プレゼンテーションタイムスタンプと、前記Playlistの再生経路を構成する前記AVストリームデータ上の指定された1つの再生区間を示す識別情報を含むことを特徴とする請求項1に記載の情報処理装置。

【請求項6】 前記ClipMarkを構成する前記マーク、または、前記PlaylistMarkを構成する前記マークは、エレメンタリーストリームのエントリーポイントを特定する情報を含むことを特徴とする請求項1に記載の情報処理装置。

【請求項7】 前記PlaylistMarkの前記マークは、ユーザが指定したお気に入りのシーンの開始点またはPlaylistのResume点を少なくとも含むタイプの情報を含むことを特徴とする請求項1に記載の情報処理装置。

【請求項8】 前記ClipMarkを構成する前記マークと前記PlaylistMarkを構成する前記マークは、前記AVストリームのエントリーポイントに対応する相対的なソースパケットのアドレスで表されることを特徴とする請求項1に記載の情報処理装置。

【請求項9】 前記ClipMarkを構成する前記マークと前記PlaylistMarkを構成する前記マークは、前記AVストリームのエントリーポイントに対応する相対的なソースパケットの第1のアドレスと、前記第1のアドレスからのオフセットのアドレスである第2のアドレスで表されることを特徴とする請求項8に記載の情報処理装置。

【請求項10】 前記第1の記録手段による記録の際に

検出された前記特徴的な画像のタイプを検出するタイプ検出手段をさらに含み、

前記第1の記録手段は、前記ClipMarkを構成する前記マークと、前記タイプ検出手段により検出された前記タイプとを対応させて記録することを特徴とする請求項1に記載の情報処理装置。

【請求項11】 前記ClipMarkの前記マークは、シーンエンジ点、コマーシャルの開始点、コマーシャルの終了点、またはタイトルが表示されたシーンを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項12】 入力されたAVストリームから抽出された特徴的な画像を指示するマークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成するとともに、

前記AVストリーム中の所定の区間の組み合わせを定義するPlaylistに対応する再生区間の中から、ユーザが任意に指定した画像を指示するマークから構成されるPlaylistMarkを生成する生成手段と、

前記ClipMark、およびPlaylistMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御手段とを有することを特徴とする情報処理方法。

【請求項13】 入力されたAVストリームから抽出された特徴的な画像を指示するマークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成するとともに、

前記AVストリーム中の所定の区間の組み合わせを定義するPlaylistに対応する再生区間の中から、ユーザが任意に指定した画像を指示するマークから構成されるPlaylistMarkを生成する生成手段と、

前記ClipMark、およびPlaylistMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御手段とを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項14】 入力されたAVストリームから抽出された特徴的な画像を指示するマークで構成されるClipMarkを、前記AVストリームを管理するための管理情報として生成するとともに、

前記AVストリーム中の所定の区間の組み合わせを定義するPlaylistに対応する再生区間の中から、ユーザが任意に指定した画像を指示するマークから構成されるPlaylistMarkを生成する生成手段と、

前記ClipMark、およびPlaylistMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御手段とをコンピュータに実行させるプログラム。

【請求項15】 AVストリームから抽出された特徴的な画像を指示するマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlaylistに対応する再生区間の中から、ユーザが任意に指定した画像を指示するマークから構成されるPlaylistMark

rkを読み出す読み出し手段と、
前記読み出し手段により読み出された前記管理情報と前記PlayListMarkによる情報を提示する提示手段と、
前記提示手段により提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照手段と、
前記参照手段により参照された前記ClipMarkを含み、前記ClipMarkに対応する位置から前記AVストリームを再生する再生手段とを含むことを特徴とする情報処理装置。

【請求項16】 前記提示手段は、前記PlayListMarkに対応するサムネイル画像によるリストをユーザに提示することを特徴とする請求項15に記載の情報処理装置。

【請求項17】 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記AVストリームのエントリポイントに対応する相対的なソースパケットのアドレスで表されることを特徴とする請求項15に記載の情報処理装置。

【請求項18】 前記ClipMarkを構成する前記マークと前記PlayListMarkを構成する前記マークは、前記AVストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、前記第1のアドレスからのオフセットのアドレスである第2のアドレスで表されることを特徴とする請求項17に記載の情報処理装置。

【請求項19】 前記ClipMarkの前記マークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、またはタイトルが表示されたシーンを含むことを特徴とする請求項15に記載の情報処理装置。

【請求項20】 AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読み出し制御ステップと、前記読み出し制御ステップの処理で読み出しが制御された前記管理情報と前記PlayListMarkによる情報を提示する提示ステップと、

前記提示ステップの処理で提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照ステップと、

前記参照ステップの処理で参照された前記ClipMarkを含み、前記ClipMarkに対応する位置からの前記AVストリームの再生を制御する再生制御ステップとを含むことを特徴とする情報処理方法。

【請求項21】 AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指

定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読み出し制御ステップと、前記読み出し制御ステップの処理で読み出しが制御された前記管理情報と前記PlayListMarkによる情報を提示する提示ステップと、
前記提示ステップの処理で提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照ステップと、
前記参照ステップの処理で参照された前記ClipMarkを含み、前記ClipMarkに対応する位置からの前記AVストリームの再生を制御する再生制御ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項22】 AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読み出し制御ステップと、前記読み出し制御ステップの処理で読み出しが制御された前記管理情報と前記PlayListMarkによる情報を提示する提示ステップと、

前記提示ステップの処理で提示された前記情報から、ユーザが再生を指示した前記PlayListに対応する前記ClipMarkを参照する参照ステップと、

前記参照ステップの処理で参照された前記ClipMarkを含み、前記ClipMarkに対応する位置からの前記AVストリームの再生を制御する再生制御ステップとをコンピュータに実行させるプログラム。

【請求項23】 AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含む前記AVストリームを管理するための管理情報と、前記AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkが、各々独立したテーブルとして記録されていることを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は情報処理装置および方法、記録媒体、プログラム、並びに記録媒体に関し、特に、AVストリームの所望の位置に、迅速にアクセスすることができるようとした情報処理装置および方法、記録媒体、プログラム、並びに記録媒体に関する。

【0002】

【従来の技術】 近年、記録可能で記録再生装置から取り外し可能なディスク型媒体として、各種の光ディスクが提案されている。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、

ビデオ信号等のAV (Audio Visual) 信号を記録するメディアとしての期待が高い。

【0003】この記録可能な光ディスクに記録するデジタルのAV信号のソース（供給源）としては、記録装置自身が、アナログ入力のオーディオビデオ信号を、MPEG-2方式で画像圧縮して作るビットストリームや、デジタルテレビジョン放送の電波から直接得られるMPEG2方式のビットストリームなどがある。一般に、デジタルテレビジョン放送では、MPEG2トランSPORTストリームが使われる。トランSPORTストリームは、トランSPORTパケットが連続したストリームであり、トランSPORTパケットは、例えば、MPEG2ビデオストリームやMPEG1オーディオストリームがパケット化されたものである。1つのトランSPORTパケットのデータ長は188バイトである。デジタルテレビジョン放送で受信されるトランSPORTストリームのAVプログラムを記録装置で光ディスクにそのまま記録すれば、ビデオやオーディオの品質を全く劣化させることなく記録することが可能である。

【0004】

【発明が解決しようとする課題】ユーザが、光ディスクに記録されているトランSPORTストリームの中から興味のあるシーン、例えば番組の頭出し点などをサーチできるようにするために、再生装置はランダムアクセス再生ができることが求められる。

【0005】一般に、MPEG2ビデオのストリームは、0.5秒程度の間隔でIピクチャを符号化し、それ以外のピクチャはPピクチャまたはBピクチャとして符号化される。したがって、MPEG2ビデオのストリームが記録された光ディスクから、ランダムアクセスし、ビデオ再生する場合、はじめに、Iピクチャをサーチしなければならない。

【0006】しかしながら、従来は、光ディスクに記録されているトランSPORTストリームに、ランダムアクセスし、ビデオ再生する場合に、Iピクチャの開始バイトを効率よくサーチすることが困難であった。すなわち、光ディスク上のトランSPORTストリームのランダムなバイト位置から、読み出したビデオストリームのシングラクスを解析し、Iピクチャの開始バイトをサーチしなければならず、Iピクチャのサーチに時間がかかり、ユーザからの入力に対して応答の速いランダムアクセス再生を行うことが困難であった。

【0007】本発明は、このような状況を鑑みてなされたものであり、ユーザのランダムアクセス再生の指示に対して、記録媒体からのトランSPORTストリームの読み出し位置の決定とストリームの復号開始を速やかに行えるようにするものである。

【0008】

【課題を解決するための手段】本発明の第1の情報処理装置は、入力されたAVストリームから抽出された特徴

的な画像を指示するマークで構成されるClipMarkを、AVストリームを管理するための管理情報として生成するとともに、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指示するマークから構成されるPlayListMarkを生成する生成手段と、ClipMark、およびPlayListMarkを各自独立したテーブルとして記録媒体に記録する記録手段とを有することを特徴とする。

【0009】前記生成手段は、ClipMarkをClipMarkInfoファイルとして生成するとともに、PlayListをPlayListファイルとして生成するようにすることができる。

【0010】前記PlayListMarkは、PlayListを再生するときのResume点を示すマークをさらに含むようにすることができる。

【0011】前記PlayListを再生するとき、PlayListの再生区間に対応するAVストリームのClipMarkを構成するマークを参照するようにすることができる。

【0012】前記PlayListMarkのマークは、プレゼンテーションタイムスタンプと、PlayListの再生経路を構成するAVストリームデータ上の指定された1つの再生区間を示す識別情報を含むようにすることができる。

【0013】前記ClipMarkを構成するマーク、または、PlayListMarkを構成するマークは、エレメンタリーストリームのエントリーポイントを特定する情報を含むようになることができる。

【0014】前記PlayListMarkのマークは、ユーザが指定したお気に入りのシーンの開始点またはPlayListのResume点を少なくとも含むタイプの情報を含むようにすることができる。

【0015】前記ClipMarkを構成するマークとPlayListMarkを構成するマークは、AVストリームのエントリーポイントに対応する相対的なソースパケットのアドレスで表されるようにすることができる。

【0016】前記ClipMarkを構成するマークとPlayListMarkを構成するマークは、AVストリームのエントリーポイントに対応する相対的なソースパケットの第1のアドレスと、第1のアドレスからのオフセットのアドレスである第2のアドレスで表されるようにすることができる。

【0017】前記第1の記録手段による記録の際に検出された特徴的な画像のタイプを検出するタイプ検出手段をさらに含み、第1の記録手段は、ClipMarkを構成するマークと、タイプ検出手段により検出されたタイプとを対応させて記録するようにすることができる。

【0018】前記ClipMarkのマークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、またはタイトルが表示されたシーンを含むようにすることができる。

【0019】本発明の第1の情報処理方法は、入力され

たAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、AVストリームを管理するための管理情報として生成するとともに、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、ClipMark、およびPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとを有することを特徴とする。

【0020】本発明の第1の記録媒体のプログラムは、入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、AVストリームを管理するための管理情報として生成するとともに、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、ClipMark、およびPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとを含むことを特徴とする。

【0021】本発明の第1のプログラムは、入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、AVストリームを管理するための管理情報として生成するとともに、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを生成する生成ステップと、ClipMark、およびPlayListMarkを各々独立したテーブルとして記録媒体に記録する際の制御を行う記録制御ステップとをコンピュータに実行させる。

【0022】本発明の第2の情報処理装置は、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むAVストリームを管理するための管理情報と、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkを読み出す読み出し手段と、読み出し手段により読み出された管理情報とPlayListMarkによる情報を提示する提示手段と、提示手段により提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照手段と、参照手段により参照されたClipMarkを含み、ClipMarkに対応する位置からAVストリームを再生する再生手段とを含むことを特徴とする。

【0023】前記提示手段は、PlayListMarkに対応するサムネイル画像によるリストをユーザに提示するようにすることができる。

【0024】前記ClipMarkを構成するマークとPlayList

Markを構成するマークは、AVストリームのエントリポイントに対応する相対的なソースパケットのアドレスで表されるようにすることができる。

【0025】前記ClipMarkを構成するマークとPlayListMarkを構成するマークは、AVストリームのエントリポイントに対応する相対的なソースパケットの第1のアドレスと、第1のアドレスからのオフセットのアドレスである第2のアドレスで表されるようにすることができる。

【0026】前記ClipMarkのマークは、シーンチェンジ点、コマーシャルの開始点、コマーシャルの終了点、またはタイトルが表示されたシーンを含むようにすることができる。

【0027】本発明の第2の情報処理装置は、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むAVストリームを管理するための管理情報と、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読み出し制御ステップと、読み出し制御ステップの処理で読み出しが制御された管理情報とPlayListMarkによる情報を提示する提示ステップと、提示ステップの処理で提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照ステップと、参照ステップの処理で参照されたClipMarkを含み、ClipMarkに対応する位置からのAVストリームの再生を制御する再生制御ステップとを含むことを特徴とする。

【0028】本発明の第2の記録媒体のプログラムは、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むAVストリームを管理するための管理情報と、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読み出し制御ステップと、読み出し制御ステップの処理で読み出しが制御された管理情報とPlayListMarkによる情報を提示する提示ステップと、提示ステップの処理で提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照ステップと、参照ステップの処理で参照されたClipMarkを含み、ClipMarkに対応する位置からのAVストリームの再生を制御する再生制御ステップとを含むことを特徴とする。

【0029】本発明のプログラムは、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むAVストリームを管理するための管理情報と、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlayListMarkの読み出しを制御する読み出し制御ステッ

と、読み出し制御ステップの処理で読み出しが制御された管理情報とPlayListによる情報を提示する提示ステップと、提示ステップの処理で提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkを参照する参照ステップと、参照ステップの処理で参照されたClipMarkを含み、ClipMarkに対応する位置からのAVストリームの再生を制御する再生制御ステップとをコンピュータに実行させる。

【0030】本発明の第3の記録媒体には、AVストリームから抽出された特徴的な画像を指示するマークで構成されるClipMarkを含むAVストリームを管理するための管理情報と、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指示するマークから構成されるPlayListMarkが、各々独立したテーブルとして記録されていることを特徴とする。

【0031】本発明の第1の情報処理装置および方法、並びにプログラムにおいては、入力されたAVストリームから抽出された特徴的な画像を指示するマークで構成されるClipMarkを、AVストリームを管理するための管理情報として生成するとともに、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指示するマークから構成されるPlayListMarkが生成され、ClipMark、およびPlayListMarkが各々独立したテーブルとして記録媒体に記録される。

【0032】本発明の第2の情報処理装置および方法、並びにプログラムは、AVストリームから抽出された特徴的な画像を指示するマークで構成されるClipMarkを含むAVストリームを管理するための管理情報と、AVストリーム中の所定の区間の組み合わせを定義するPlayListに対応する再生区間の中から、ユーザが任意に指定した画像を指示するマークから構成されるPlayListMarkが読み出され、その読み出された管理情報とPlayListMarkによる情報が提示され、提示された情報から、ユーザが再生を指示したPlayListに対応するClipMarkが参照され、参照されたClipMarkを含み、ClipMarkに対応する位置からAVストリームが再生される。

【0033】

【発明の実施の形態】以下に、本発明の実施の形態について、図面を参照して説明する。図1は、本発明を適用した記録再生装置1の内部構成例を示す図である。まず、外部から入力された信号を記録媒体に記録する動作を行う記録部2の構成について説明する。記録再生装置1は、アナログデータ、または、デジタルデータを入力し、記録することができる構成とされている。

【0034】端子11には、アナログのビデオ信号が、端子12には、アナログのオーディオ信号が、それぞれ入力される。端子11に入力されたビデオ信号は、解析部14とAVエンコーダ15に、それぞれ出力される。端

子12に入力されたオーディオ信号は、解析部14とAVエンコーダ15に出力される。解析部14は、入力されたビデオ信号とオーディオ信号からシーンチェンジなどの特徴点を抽出する。

【0035】AVエンコーダ15は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、およびAV同期等のシステム情報(S)をマルチプレクサ16に出力する。

【0036】符号化ビデオストリームは、例えば、MPEG(Moving Picture Expert Group)2方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG1方式により符号化されたオーディオストリームや、ドルビーAC3方式(商標)により符号化されたオーディオストリーム等である。マルチプレクサ16は、入力されたビデオおよびオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ17を介して多重化ストリーム解析部18とソースパケッタイザ19に出力する。

【0037】多重化ストリームは、例えば、MPEG2トランスポートストリームやMPEG2プログラムストリームである。ソースパケッタイザ19は、入力された多重化ストリームを、そのストリームを記録させる記録媒体100のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームに符号化する。AVストリームは、ECC(誤り訂正)符号化部20と変調部21でECC符号の付加と変調処理が施され、書き込み部22に出力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100にAVストリームファイルを書き込む(記録する)。

【0038】デジタルインタフェースまたはデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子13に入力される。端子13に入力されたトランスポートストリームの記録方式には、2通りあり、それらは、トランスペアレントに記録する方式と、記録ビットレートを下げるなどの目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインターフェースとしての端子24から制御部23へ入力される。

【0039】入力トランスポートストリームをトランスペアレントに記録する場合、端子13に入力されたトランスポートストリームは、スイッチ17を介して多重化ストリーム解析部18と、ソースパケッタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0040】入力トランスポートストリームを再エンコードした後に記録する場合、端子13に入力されたトランスポートストリームは、デマルチプレクサ26に入力

される。デマルチプレクサ26は、入力されたトランSPORTストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)を抽出する。

【0041】デマルチプレクサ26により抽出されたストリーム(情報)のうち、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチプレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ16に出力する。

【0042】一方、デマルチプレクサ26から出力され、マルチプレクサ16に入力されたオーディオストリームとシステム情報、および、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部18とソースパケットタイザ19にスイッチ17を介して出力される。これ以後の記録媒体100へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0043】本実施の形態の記録再生装置1は、AVストリームのファイルを記録媒体100に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部23により作成される。制御部23への入力情報は、解析部14からの動画像の特徴情報、多重化ストリーム解析部18からのAVストリームの特徴情報、および端子24から入力されるユーザからの指示情報である。

【0044】解析部14から供給される動画像の特徴情報は、AVエンコーダ15がビデオ信号を符号化する場合において、解析部14により生成されるものである。解析部14は、入力ビデオ信号とオーディオ信号の内容を解析し、入力動画像信号の中の特徴的な画像(クリップマーク)に関する情報を生成する。これは、例えば、入力ビデオ信号の中のプログラムの開始点、シーンエンジ点やCMコマーシャルのスタート点・エンド点、タイトルやテロップなどの特徴的なクリップマーク点の画像の指示情報であり、また、それにはその画像のサムネールも含まれる。さらにオーディオ信号のステレオとモノラルの切り替え点や、無音区間などの情報も含まれる。

【0045】これらの画像の指示情報は、制御部23を介して、マルチプレクサ16へ入力される。マルチプレクサ16は、制御部23からクリップマークとして指定される符号化ピクチャを多重化する時に、その符号化ピクチャをAVストリーム上で特定するための情報を制御部23に返す。具体的には、この情報は、ピクチャのPT

S(プレゼンテーションタイムスタンプ)またはその符号化ピクチャのAVストリーム上でのアドレス情報である。制御部23は、特徴的な画像の種類とその符号化ピクチャをAVストリーム上で特定するための情報を関連付けて記憶する。

【0046】多重化ストリーム解析部18からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関する情報であり、解析部18により生成される。例えば、AVストリーム内の1ピクチャのタイムスタンプとアドレス情報、システムタイムクロックの不連続点情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報などが含まれる。また、端子13から入力されるトランSPORTストリームをトランスペアレン特徴化して記録する場合、多重化ストリーム解析部18は、入力トランSPORTストリームの中から前出のクリップマークの画像を検出し、その種類とクリップマークで指定するピクチャを特定するための情報を生成する。

【0047】端子24からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリリューム点の情報などである。

【0048】制御部23は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベース、記録媒体100の記録内容の管理情報(info.dvr)、およびサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様にして、ECC符号化部20、変調部21で処理されて、書き込み部22へ入力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100へデータベースファイルを記録する。

【0049】上述したアプリケーションデータベース情報についての詳細は後述する。

【0050】このようにして記録媒体100に記録されたAVストリームファイル(画像データと音声データのファイル)と、アプリケーションデータベース情報が再生部3により再生される場合、まず、制御部23は、読み出し部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読み出し部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部29とECC復号部30の復調と誤り訂正処理を経て、制御部23へ入力される。

【0051】制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインターフェースへ

出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しが、読み出し部28に指示する。読み出し部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、所定の処理が施されることにより復調され、さらにECC復号部30の処理を経て、ソースデパケッタイザ31出力される。

【0052】ソースデパケッタイザ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ26が処理可能なストリームに変換する。デマルチプレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、およびAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

【0053】また、ユーザインタフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記録媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しが、読み出し部28に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部23は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読み出し部28に指示する。

【0054】また、Clip Informationの中のClipMarkにストアされている番組の頭出し点やシーンチェンジ点の中から、ユーザがあるクリップマークを選択した時(例えば、この動作は、ClipMarkにストアされている番組の頭出し点やシーンチェンジ点のサムネール画像リストをユーザーインタフェースに表示して、ユーザが、その中からある画像を選択することにより行われる)、制御部23は、Clip Informationの内容に基づいて、記録媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しが読み出し部28へ指示する。すなわち、ユーザが選択した画像がストアされているAVストリーム上でのアドレスに最も近いアドレスにあるIピクチャからのデータを読み出すように読み出し部28へ指示する。読み出し部28は、指定されたアドレスからデータを読み出し、読み出されたデータは、復調部29、ECC復号部30、ソースデパケッタイザ31の処理を経て、デマルチプレクサ26へ入力され、AVデコーダ27で復号されて、マーク点のピクチャのアドレスで示されるAVデータが再生される。

【0055】また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読み出し部28に指示する。

【0056】読み出し部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

【0057】次に、ユーザが、記録媒体100に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を再生し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子24から再生区間の開始点(イン点)と終了点(アウト点)の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成する。

【0058】ユーザが、記録媒体100に記録されているAVストリームの一部を消去したい場合、ユーザインタフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不要なストリーム部分を消去するように、書き込み部22に指示する。

【0059】ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、かつ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成し、さらに、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

【0060】まず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読み出し部28にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しが指示する。そして、読み出し部28は、記録媒体100からデータを読み出し、そのデータは、復調部29、ECC復号部30、ソースデパケッタイザ31を経て、デマルチプレクサ26に出力される。

【0061】制御部23は、デマルチプレクサ26に入力されたデータを解析して、ビデオストリームの再エンコード方法(picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て)と、再多重化方式を

決定し、その方式をAVエンコーダ15とマルチプレクサ16に供給する。

【0062】次に、デマルチプレクサ26は、入力されたストリームをビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)に分離する。ビデオストリームは、AVデコーダ27に入力されるデータとマルチプレクサ16に入力されるデータがある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ27で復号され、復号されたピクチャはAVエンコーダ15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ16に入力される。

【0063】マルチプレクサ16は、制御部23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、EC符号化部20、変調部21で処理されて、書き込み部22に入力される。書き込み部22は、制御部23から供給される制御信号に基づいて、記録媒体100にAVストリームを記録する。

【0064】以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図2は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlaylistとClipの2つのレイヤをもつ。Volume Informationは、ディスク内のすべてのClipとPlaylistの管理をする。ここでは、1つのAVストリームとその付属情報のペアを1つのオブジェクトと考え、それをClipと称する。AVストリームファイルはClip AV stream fileと称し、その付属情報は、Clip Information fileと称する。

【0065】1つのClip AV stream fileは、MPEG2トランスポートストリームをアプリケーションフォーマットによって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリーポイント(ピクチャ)は、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられた時、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

【0066】Playlistについて、図3を参照して説明する。Playlistは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようになるために設けられている。1つのPlaylistは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイン点(IN)とアウト点(OUT)の対で表される。従って、P

layListは、複数のPlayItemが集まることにより構成される。

【0067】Playlistには、2つのタイプがある。1つは、Real Playlistであり、もう1つは、Virtual Playlistである。Real Playlistは、それが参照しているClipのストリーム部分を共有している。すなわち、Real Playlistは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real Playlistが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

【0068】Virtual Playlistは、Clipのデータを共有していない。従って、Virtual Playlistが変更または消去されたとしても、Clipの内容には何も変化が生じない。

【0069】次に、Real Playlistの編集について説明する。図4(A)は、Real Playlistのクリエイト(create:作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal Playlistが新たに作成される操作である。

【0070】図4(B)は、Real Playlistのディバイド(divide:分割)に関する図であり、Real Playlistが所望な点で分けられて、2つのReal Playlistに分割される操作である。この分割という操作は、例えば、1つのPlaylistにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番組として登録(記録)し直したいといったようなときに行われる。この操作により、Clipの内容が変更される(Clip自体が分割される)ことはない。

【0071】図4(C)は、Real Playlistのコンバイン(combine:結合)に関する図であり、2つのReal Playlistを結合して、1つの新しいReal Playlistにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される(Clip自体が1つにされる)ことはない。

【0072】図5(A)は、Real Playlist全体のデリート(delete:削除)に関する図であり、所定のReal Playlist全体を消去する操作がされた場合、削除されたReal Playlistが参照するClipの、対応するストリーム部分も削除される。

【0073】図5(B)は、Real Playlistの部分的な削除に関する図であり、Real Playlistの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

【0074】図5(C)は、Real Playlistのミニマイズ(minimize:最小化)に関する図であり、Real Playlistに対応するPlayItemを、Virtual Playlistに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual Playlistにとって不必要的Clipの、対応

するストリーム部分は削除される。

【0075】上述したような操作により、Real Playlistが変更されて、そのReal Playlistが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual Playlistが存在し、そのVirtual Playlistにおいて、削除されたClipにより問題が生じる可能性がある。

【0076】そのようなことが生じないように、ユーザに、削除という操作に対して、「そのReal Playlistが参照しているClipのストリーム部分を参照しているVirtual Playlistが存在し、もし、そのReal Playlistが消去されると、そのVirtual Playlistもまた消去されることになるが、それでも良いか？」といったメッセージなどを表示させることにより、確認（警告）を促した後に、ユーザの指示により削除の処理を実行、または、キャンセルする。または、Virtual Playlistを削除する代わりに、Real Playlistに対してミニマイズの操作が行われるようにする。

【0077】次にVirtual Playlistに対する操作について説明する。Virtual Playlistに対して操作が行われたとしても、Clipの内容が変更されることはない。図6は、アセンブル(Assemble)編集(IN-OUT編集)に関する図であり、ユーザが見たいと所望した再生区間のPlay Itemを作り、Virtual Playlistを作成するといった操作である。Play Item間のシームレス接続が、アプリケーションフォーマットによりサポートされている（後述）。

【0078】図6(A)に示したように、2つのReal Playlist1、2と、それぞれのReal Playlistに対応するClip1、2が存在している場合に、ユーザがReal Playlist1内の所定の区間(In1乃至Out1までの区間: PlayItem1)を再生区間として指示し、続けて再生する区間として、Real Playlist2内の所定の区間(In2乃至Out2までの区間: PlayItem2)を再生区間として指示したとき、図6(B)に示すように、PlayItem1とPlayItem2から構成される1つのVirtual Playlistが作成される。

【0079】次に、Virtual Playlistの再編集(Re-editing)について説明する。再編集には、Virtual Playlistの中のイン点やアウト点の変更、Virtual Playlistへの新しいPlay Itemの挿入(insert)や追加 append)、Virtual Playlistの中のPlay Itemの削除などがある。また、Virtual Playlistそのものを削除することもできる。

【0080】図7は、Virtual Playlistへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual Playlistへのオーディオのアフレコをサブパスとして登録することである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlaylistのメインパスのAVストリームに、付加的なオーディオストリームが、サブパスとして付加される。

【0081】Real PlaylistとVirtual Playlistで共通の操作として、図8に示すようなPlaylistの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlaylistの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of Playlist(図20などを参照して後述する)によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

【0082】次に、マーク(Mark)について説明する。マークは、図9に示されるように、ClipおよびPlaylistの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、ClipMark(クリップマーク)と呼ばれる。ClipMarkは、AVストリームの内容に起因する特徴的なシーンを指定する、例えば番組の頭だし点やシーンチェンジ点などである。ClipMarkは、図1の例えば解析部14によって生成される。Playlistを再生する時、そのPlaylistが参照するClipのマークを参照して、使用することができる。

【0083】Playlistに付加されるマークは、Playlist Mark(プレイリストマーク)と呼ばれる。Playlist Markは、主にユーザによってセットされる、例えば、ブックマークやリリューム点などである。ClipまたはPlaylistにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【0084】ClipMarkの別のフォーマットとして、Clip Markが参照するピクチャをAVストリームの中でのアドレスベースで指定するようにしても良い。Clipにマークをセットすることは、マーク点のピクチャを示すアドレスベースの情報をマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマーク点のピクチャを示すアドレスベースの情報を除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【0085】次にサムネイルについて説明する。サムネイルは、Volume、Playlist、およびClipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル(不図示)などを操作して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

【0086】Volumeと各Playlistは代表画を持つことができるようになる必要がある。Volumeの代表画は、ディスク(記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクと記述する)を記録再生装置1の所定の場所にセットした時に、そのデ

ィスクの内容を表す静止画を最初に表示する場合などに用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを想定している。

【0087】Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上Volumeを表す代表画としてのサムネイルと、Playlistを表す代表画としてのサムネイルの2種類のサムネイルをメニューサムネイルと称する。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、すべてのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図10に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込こまれた画像でもよい。

【0088】一方、ClipとPlaylistには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることが出来るようになる必要がある。このようなマーク点を表すピクチャをマークサムネイル（Mark Thumbnails）と称する。従って、マークサムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

【0089】図11は、Playlistに付けられるマークと、そのマークサムネイルの関係について示す図であり、図12は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表す時に、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置1がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかるても、問題にはならない。

【0090】また、ボリューム内に存在するファイル数を減らすために、すべてのマークサムネイルは1つのファイルに格納するのがよい。Playlistはメニューサムネイル1つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

【0091】図13は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、Playlist、およびClipの関係について示した図である。メニューサムネイルファイルには、Playlist毎に設けられたメニューサムネイルがファイルされている。メニューサムネイル

ファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各Playlist毎と各Clip毎に作成されたサムネイルがファイルされている。

【0092】次に、CPI（Characteristic Point Information）について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられた時、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本実施の形態では、2種類のCPIを用いる。1つは、EP_mapであり、もう一つは、TU_mapである。

【0093】EP_mapは、エントリーポイント（EP）データのリストであり、それはエレメンタリーストリームおよびトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリーポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ（PTS）と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

【0094】EP_mapは、主に2つの目的のために使用される。第1に、Playlistの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができるとき、EP_mapが作成され、ディスクに記録される。

【0095】TU_mapは、デジタルインターフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット（TU）データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU_mapが作成され、ディスクに記録される。

【0096】STCInfoは、MPEG2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。仮に、AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れる可能性がある。そのため、AVストリーム上の所定の時刻をPTSベースで指す場合、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。

【0097】更に、そのPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットでは、STC-sequenceと呼び、そのインデックスをSTC-sequence-idと記述する。STC-sequenceの

情報は、Clip Information fileのSTCInfoで定義される。STC-sequence-idは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

【0098】プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ1つのシステムタイムベースを共有するものである。再生装置にとって、AVストリームのデコードに先だち、そのAVストリームの内容がわかることは有用である。例えば、ビデオやオーディオのエレメンタリストリームを伝送するトランスポートパケットのPIDの値や、ビデオやオーディオのコンポーネント種類（例えば、HDTVのビデオとMPEG-2 AACのオーディオストリームなど）などの情報である。

【0099】この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するところのメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だって、再生装置のAVデコーダおよびデマルチプレクサの初期状態をセットするために役立つ。この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

【0100】MPEG2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリストリームを伝送するところのトランスポートパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化するなどである。

【0101】ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をProgram-sequenceと呼ぶ。Program-sequenceは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

【0102】本実施の形態では、セルフエンコードのストリームフォーマット（SESF）を定義する。SESFは、アナログ入力信号を符号化する目的、およびデジタル入力信号（例えばDV）をデコードしてからMPEG2トランスポートストリームに符号化する場合に用いられる。

【0103】SESFは、MPEG-2トランスポートストリームおよびAVストリームについてのエレメンタリストリームの符号化制限を定義する。記録再生装置1が、SESFストリームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

【0104】デジタル放送のストリームは、次に示す方式のうちのいずれかが用いられて記録媒体100に記録される。まず、デジタル放送のストリームをSESFストリームにトランスポーティングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この

場合、EP_mapが作成されて、ディスクに記録されなければならない。

【0105】あるいは、デジタル放送ストリームを構成するエレメンタリストリームを新しいエレメンタリストリームにトランスポーティングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【0106】例えば、入力ストリームがISDB（日本のデジタルBS放送の規格名称）準拠のMPEG-2トランスポートストリームであり、それがHDTVビデオストリームとMPEG AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスポーティングし、そのSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトランスポートストリームは、共にISDBフォーマットに準拠しなければならない。

【0107】デジタル放送のストリームが、記録媒体100に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にEP_mapが作成されてディスクに記録される。

【0108】または、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にTU_mapが作成されてディスクに記録される。

【0109】次にディレクトリとファイルについて説明する。以下、記録再生装置1をDVR（Digital Video Recording）と適宜記述する。図14はディスク上のディレクトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図14に示したように、“DVR”ディレクトリを含むrootディレクトリ、“PLAYLIST”ディレクトリ、“CLIPINF”ディレクトリ、“M2TS”ディレクトリ、および“DATA”ディレクトリを含む“DVR”ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしても良いが、それらは、本実施の形態のアプリケーションフォーマットでは、無視されるとする。

【0110】“DVR”ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。“DVR”ディレクトリは、4個のディレクトリを含む。“PLAYLIST”ディレクトリの下には、Real PlaylistとVirtual Playlistのデータベースファイルが置かれる。このディレクトリは、Playlistが1つもなくても存在する。

【0111】“CLIPINF”ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。“M2TS”ディレクトリの下に

は、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。“DATA”ディレクトリは、デジタルTV放送などのデータ放送のファイルがストアされる。

【0112】“DVR”ディレクトリは、次に示すファイルをストアする。“info.dvr”ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ一つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されるとする。“menu.thmb”ファイルは、メニュー・サムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは1つのメニュー・サムネイルがなければならない。ファイル名は、menu.thmbに固定されるとする。メニュー・サムネイル画像が1つもない場合、このファイルは、存在しなくても良い。

【0113】“mark.thmb”ファイルは、マーク・サムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは1つのマーク・サムネイルがなければならない。ファイル名は、mark.thmbに固定されるとする。メニュー・サムネイル画像が1つもない場合、このファイルは、存在しなくても良い。

【0114】“PLAYLIST”ディレクトリは、2種類のPlaylistファイルをストアするものであり、それらは、Real PlaylistとVirtual Playlistである。“xxxxx.rpls”ファイルは、1つのReal Playlistに関連する情報をストアする。それぞれのReal Playlist毎に、1つのファイルが作られる。ファイル名は、“xxxxx.rpls”である。ここで、“xxxxx”は、5個の0乃至9までの数字である。ファイル拡張子は、“rpls”でなければならないとする。

【0115】“yyyyy.vpls”ファイルは、1つのVirtual Playlistに関連する情報をストアする。それぞれのVirtual Playlist毎に、1つのファイルが作られる。ファイル名は、“yyyyy.vpls”である。ここで、“yyyyy”は、5個の0乃至9までの数字である。ファイル拡張子は、“vpls”でなければならないとする。

【0116】“CLIPINF”ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。“zzzzz.clpi”ファイルは、1つのAVストリームファイル(Clip AV stream fileまたはBridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、“zzzzz.clpi”であり、“zzzzz”は、5個の0乃至9までの数字である。ファイル拡張子は、“clpi”でなければならないとする。

【0117】“M2TS”ディレクトリは、AVストリームのファイルをストアする。“zzzzz.m2ts”ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream fileまたはBridge-Clip AV streamである。ファイル名は、“zzzzz.m2ts”であり、“zzzzz”は、5個の0乃至9までの数字である。ファイル拡張子は、“m2ts”でなければならないとする。

【0118】“DATA”ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイルなどである。

【0119】次に、各ディレクトリ(ファイル)のシンタクスとセマンティクスを説明する。まず、“info.dvr”ファイルについて説明する。図15は、“info.dvr”ファイルのシンタクスを示す図である。“info.dvr”ファイルは、3個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、およびMakersPrivateData()である。

【0120】図15に示したinfo.dvrのシンタクスについて説明するに、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlaylist()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0121】MakersPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word(パディングワード)は、info.dvrのシンタクスに従って挿入される。N1とN2は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【0122】DVRVolume()は、ボリューム(ディスク)の内容を記述する情報をストアする。図16は、DVRVolume()のシンタクスを示す図である。図16に示したDVRVolume()のシンタクスを説明するに、version_numberは、このDVRVolume()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、“0045”と符号化される。

【0123】lengthは、このlengthフィールドの直後からDVRVolume()の最後までのDVRVolume()のバイト数を示す32ビットの符号なし整数で表される。

【0124】ResumeVolume()は、ボリュームの中で最後に再生したReal PlaylistまたはVirtual Playlistのファイル名を記憶している。ただし、Real PlaylistまたはVirtual Playlistの再生をユーザが中断した時の再生位置は、PlaylistMark()において定義されるresume_markにストアされる(図42、図43)。

【0125】図17は、ResumeVolume()のシンタクスを示す図である。図17に示したResumeVolume()のシンタクスを説明するに、valid_flagは、この1ビットのフラグが1にセットされている場合、resume_PlayList_nameフィールドが有効であることを示し、このフラグが0にセットされている場合、resume_PlayList_nameフィールドが無効であることを示す。

【0126】resume_PlayList_nameの10バイトのフィールドは、リジュームされるべきReal PlaylistまたはVirtual Playlistのファイル名を示す。

【0127】図16に示したDVRVolume()のシンタクス

のなかの、UIAppInfoVolume は、ボリュームについてのユーザインターフェースアプリケーションのパラメータをストアする。図18は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティクスを説明するに、character_setの8ビットのフィールドは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

【0128】name_lengthの8ビットフィールドは、Volume_nameフィールドの中に示されるボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0129】Volume_protect_flagは、ボリューム中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せる事（再生される事）が許可される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくとも、そのボリュームのコンテンツを、ユーザに見せる事が許可される。

【0130】最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが0にセットされているか、または、このフラグが1にセットされてもユーザがPIN番号を正しく入力できたならば、記録再生装置1は、そのディスクの中のPlayListの一覧を表示させる。それぞれのPlayListの再生制限は、volume_protect_flagとは無関係であり、それはUIAppInfoPlayList()の中に定義されるplayback_control_flagによって示される。

【0131】PINは、4個の0乃至9までの数字で構成され、それぞれの数字は、ISO/IEC 646に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

【0132】次に図15に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、PlayList(Real PlayListとVirtual PlayList)のファイル名をストアする。ボリュームに記録され

ているすべてのPlayListファイルは、TableOfPlayLists()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlayListのデフォルトの再生順序を示す。

【0133】図20は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明するに、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバーを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0134】lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す32ビットの符号なしの整数である。number_of_PlayListsの16ビットのフィールドは、PlayList_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlayListの数に等しくなければならない。PlayList_file_nameの10バイトの数字は、PlayListのファイル名を示す。

【0135】図21は、TableOfPlayLists()のシンタクスの別の構成を示す図である。図21に示したシンタクスは、図20に示したシンタクスに、UIAppinfoPlayList（後述）を含ませた構成とされている。このように、UIAppinfoPlayListを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図20に示したシンタクスを用いるとして以下の説明をする。

【0136】図15に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置1のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、それを定義したメーカーを識別するために標準化されたmaker_IDを持つ。MakersPrivateData()は、1つ以上のmaker_IDを含んでも良い。

【0137】所定のメーカーが、プライベートデータを挿入したい時に、すでに他のメーカーのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカーは、既にある古いプライベートデータを消去するではなく、新しいプライベートデータをMakersPrivateData()の中に追加するようにする。このように、本実施の形態においては、複数のメーカーのプライベートデータが、1つのMakersPrivateData()に含まれることが可能であるようにする。

【0138】図22は、MakersPrivateDataのシンタクスを示す図である。図22に示したMakersPrivateDataのシンタクスについて説明するに、version_numberは、このMakersPrivateData()のバージョンナンバーを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPriva

teData()の最後までのMakersPrivateData()のバイト数を示す32ビットの符号なし整数を示す。

【0139】mpd_blocks_start_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd_block()の先頭バイトアドレスを示す。相対バイト数はゼロからカウントされる。number_of_maker_entriesは、MakersPrivateData()の中に含まれているメーカプライベートデータのエントリー数を与える16ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker_IDの値を持つメーカプライベートデータが2個以上存在してはならない。

【0140】mpd_block_sizeは、1024バイトを単位として、1つのmpd_blockの大きさを与える16ビットの符号なし整数である。例えば、mpd_block_size=1ならば、それは1つのmpd_blockの大きさが1024バイトであることを示す。number_of_mpd_blocksは、MakersPrivateData()の中に含まれるmpd_blockの数を与える16ビットの符号なし整数である。maker_IDは、そのメーカプライベートデータを作成したDVRシステムの製造メーカを示す16ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンサによって指定される。

【0141】maker_model_codeは、そのメーカプライベートデータを作成したDVRシステムのモデルナンバーコードを示す16ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカによって設定される。start_mpd_block_numberは、そのメーカプライベートデータが開始されるmpd_blockの番号を示す16ビットの符号なし整数である。メーカプライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

【0142】mpd_lengthは、バイト単位でメーカプライベートデータの大きさを示す32ビットの符号なし整数である。mpd_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中のすべてのmpd_blockは、同じサイズでなければならない。

【0143】次に、Real Playlist fileとVirtual Playlist fileについて、換言すれば、xxxxx.rplsとyyyyy.vplsについて説明する。図23は、xxxxx.rpls (Real Playlist)、または、yyyyy.vpls (Virtual Playlist)のシンタクスを示す図である。xxxxx.rplsとyyyyy.vplsは、同一のシンタクス構成をもつ。xxxxx.rplsとyyyyy.vplsは、それぞれ、3個のオブジェクトから構成され、それらは、Playlist()、PlaylistMark()、およびMakersPrivateData()である。

【0144】PlaylistMark_Start_addressは、Playlistファイルの先頭のバイトからの相対バイト数を単位として、PlaylistMark()の先頭アドレスを示す。相対バイト

数はゼロからカウントされる。

【0145】MakersPrivateData_Start_addressは、Playlistファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0146】padding_word (パディングワード) は、Playlistファイルのシンタクスにしたがって挿入され、N1とN2は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにも良い。

【0147】ここで、既に、簡便に説明したが、Playlistについてさらに説明する。ディスク内にあるすべてのReal Playlistによって、Bridge-Clip (後述) を除くすべてのClipの中の再生区間が参照されていなければならない。かつ、2つ以上のReal Playlistが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

【0148】図24を参照してさらに説明するに、図24 (A) に示したように、全てのClipは、対応するReal Playlistが存在する。この規則は、図24 (B) に示したように、編集作業が行われた後においても守られる。従って、全てのClipは、どれかしらのReal Playlistを参照することにより、必ず視聴することが可能である。

【0149】図24 (C) に示したように、Virtual Playlistの再生区間は、Real Playlistの再生区間またはBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual Playlistにも参照されないBridge-Clipがディスクの中に存在してはならない。

【0150】Real Playlistは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual Playlistは、PlayItemのリストを含み、Playlist()の中に示されるClip_typeがEP_map_typeであり、かつPlaylist_typeが0 (ビデオとオーディオを含むPlaylist) である場合、Virtual Playlistは、ひとつのSubPlayItemを含む事ができる。本実施の形態におけるPlaylist()では、SubPlayItemはオーディオのアフレコの目的にだけに使用される、そして、1つのVirtual Playlistが持つSubPlayItemの数は、0または1でなければならない。

【0151】次に、Playlistについて説明する。図25は、Playlistのシンタクスを示す図である。図25に示したPlaylistのシンタクスを説明するに、version_numberは、このPlaylist()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からPlaylist()の最後までのPlaylist()のバイト数を示す32ビットの符号なし整数である。Playlist_typeは、このPlaylistのタイプを示す8ビットのフィールドであり、その一例を図26に示す。

【0152】CPI_typeは、1ビットのフラグであり、PlayItem()およびSubPlayItem()によって参照されるClipのCPI_typeの値を示す。1つのPlayListによって参照される全てのClipは、それらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、PlayListの中にあるPlayItemの数を示す16ビットのフィールドである。

【0153】所定のPlayItem()に対応するPlayItem_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem_idは、0から開始される。number_of_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す16ビットのフィールドである。この値は、0または1である。付加的なオーディオストリームのパス(オーディオストリームパス)は、サブパスの一種である。

【0154】次に、図25に示したPlayListのシンタクスのUIAppInfoPlayListについて説明する。UIAppInfoPlayListは、PlayListについてのユーザインターフェースアプリケーションのパラメータをストアする。図27は、UIAppInfoPlayListのシンタクスを示す図である。図27に示したUIAppInfoPlayListのシンタクスを説明するに、character_setは、8ビットのフィールドであり、PlayList_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示したテーブルに準拠する値に対応する。

【0155】name_lengthは、8ビットフィールドであり、PlayList_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。PlayList_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0156】record_time_and_dateは、PlayListが記録された時の日時をストアする56ビットのフィールドである。このフィールドは、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、2001/12/23:01:02:03は、“0x20011223010203”と符号化される。

【0157】durationは、PlayListの総再生時間を時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、01:45:30は、“0x014530”と符号化される。

【0158】valid_periodは、PlayListが有効である期間を示す32ビットのフィールドである。このフィールドは、8個の数字を4ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、記録再生装置1は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07は、“0x

20010507”と符号化される。

【0159】maker_idは、そのPlayListを最後に更新したDVRプレーヤ(記録再生装置1)の製造者を示す16ビットの符号なし整数である。maker_idに符号化される値は、DVRフォーマットのライセンサによって割り当てる。maker_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す16ビットの符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

【0160】playback_control_flagのフラグが1にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

【0161】write_protect_flagは、図28(A)にテーブルを示すように、1にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去および変更されない。このフラグが0にセットされている場合、ユーザは、そのPlayListを自由に消去および変更できる。このフラグが1にセットされている場合、ユーザが、そのPlayListを消去、編集、または上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

【0162】write_protect_flagが0にセットされているReal PlayListが存在し、かつ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが1にセットされても良い。ユーザが、Real PlayListを消去しようとする場合、記録再生装置1は、そのReal PlayListを消去する前に、上記Virtual PlayListの存在をユーザに警告するか、または、そのReal PlayListを“Minimize”する。

【0163】is_played_flagは、図28(B)に示すように、フラグが1にセットされている場合、そのPlayListは、記録されてから一度は再生されたことを示し、0にセットされている場合、そのPlayListは、記録されてから一度も再生されたことを示す。

【0164】archiveは、図28(C)に示すように、そのPlayListがオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。ref_thumbnail_indexのフィールドは、PlayListを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのPlayListには、PlayListを代表するサムネイル画像が付加さ

れていない。

【0165】次にPlayItemについて説明する。1つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_information_file_name、Clipの再生区間を特定するためのIN_timeとOUT_timeのペア、PlayList()において定義されるCPI_typeがEP_map_typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、および、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection_conditionである。

【0166】PlayListが2つ以上のPlayItemから構成される時、それらのPlayItemはPlayListのグローバル時間軸上に、時間のギャップまたはオーバーラップなしに一列に並べられる。PlayList()において定義されるCPI_typeがEP_map_typeであり、かつ現在のPlayItemがBridgeSequence()を持つない時、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じSTC連続区間上の時間を指していくなければならない。そのような例を図29に示す。

【0167】図30は、PlayList()において定義されるCPI_typeがEP_map_typeであり、かつ現在のPlayItemがBridgeSequence()を持つ時、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time(図の中でIN_time1と示されているもの)は、先行するPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT_time(図の中でOUT_time1と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

【0168】現在のPlayItemのIN_time(図の中でIN_time2と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのPlayItemのOUT_time(図の中でOUT_time2と示されているもの)は、現在のPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。

【0169】図31に示すように、PlayList()のCPI_typeがTU_map_typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AVストリーム上の時間を指している。

【0170】PlayItemのシンタクスは、図32に示すようになる。図32に示したPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、ClipInformation_fileのファイル名を示す。このClipInformation_fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

【0171】STC_sequence_idは、8ビットのフィールドであり、PlayItemが参照するSTC連続区間のSTC_sequence_idを示す。PlayList()の中で指定されるCPI_typeがTU_map_typeである場合、この8ビットフィールドは何も意味を持たず、0にセットされる。IN_timeは、32ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図33に示すように、PlayList()において定義されるCPI_typeによって異なる。

【0172】OUT_timeは、32ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図34に示すように、PlayList()において定義されるCPI_typeによって異なる。

【0173】Connection_Conditionは、図35に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す2ビットのフィールドである。図36は、図35に示したConnection_Conditionの各状態について説明する図である。

【0174】次に、BridgeSequenceInfoについて、図37を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information file(図45)を指定するBridge_Clip_Information_file_nameを含む。

【0175】また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN_exit_from_previous_Clipと称される。さらに現在のPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN_enter_to_current_Clipと称される。

【0176】図37において、RSPN_arrival_time_disccontinuityは、the Bridge-Clip AVstreamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、ClipInfo()(図46)において定義される。

【0177】図38は、BridgeSequenceinfoのシンタクスを示す図である。図38に示したBridgeSequenceinfoのシンタクスを説明するに、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、「Bridge-Clip AV stream」を示していなければならない。

【0178】RSPN_exit_from_previous_Clipの32ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、この

ソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【0179】RSPN_enter_to_current_Clipの32ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【0180】次に、SubPlayItemについて、図39を参照して説明する。SubPlayItem()の使用は、PlayList()のCPI_typeがEP_map_typeである場合だけに許される。本実施の形態においては、SubPlayItemはオーディオのアフレコの目的のためだけに使用されるとする。SubPlayItem()は、次に示すデータを含む。まず、PlayListの中のsub pathが参照するClipを指定するためのClip_information_file_nameを含む。

【0181】また、Clipの中のsub pathの再生区間を指定するためのSubPath_IN_timeとSubPath_OUT_timeを含む。さらに、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync_PlayItem_idとsync_start PTS_of_PlayItemを含む。sub pathに参照されるオーディオのClip AV streamは、STC不連続点（システムタイムベースの不連続点）を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

【0182】図40は、SubPlayItemのシンタクスを示す図である。図40に示したSubPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip_Information_fileのファイル名を示し、それはPlayListの中でsub pathによって使用される。このClip_Information_fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならぬ。

【0183】SubPath_typeの8ビットのフィールドは、sub pathのタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

【0184】sync_PlayItem_idの8ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻が含まれるPlayItemのPlayItem_idを示す。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図25参照）。

PlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図25参照）。

【0185】sync_start PTS_of_PlayItemの32ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻を示し、sync_PlayItem_idで参照されるPlayItem上のPTS（Presentation Time Stamp）の上位32ビットを示す。SubPath_IN_timeの32ビットフィールドは、Sub pathの再生開始時刻をストアする。SubPath_IN_timeは、Sub Pathの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

【0186】SubPath_OUT_timeの32ビットフィールドは、Sub pathの再生終了時刻をストアする。SubPath_OUT_timeは、次式によって算出されるPresentation_end_TSの値の上位32ビットを示す。Presentation_end_TS = PTS_out + AU_durationここで、PTS_outは、SubPathの最後のプレゼンテーションユニットに対応する33ビット長のPTSである。AU_durationは、SubPathの最後のプレゼンテーションユニットの90kHz単位の表示期間である。

【0187】次に、図23に示したxxxxx.rplsとyyyyy.vplsのシンタクス内のPlayListMark()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図42は、PlayListMarkのシンタクスを示す図である。図42に示したPlayListMarkのシンタクスについて説明するに、version_numberは、このPlayListMark()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従つて、"0045"と符号化されなければならない。

【0188】lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト数を示す32ビットの符号なし整数である。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークの個数を示す16ビットの符号なし整数である。number_of_PlayList_marksは、0であってよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図43に示すテーブルに従って符号化される。

【0189】mark_time_stampの32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図44に示すように、PlayList()において定義されるCPI_typeによって異なる。PlayItem_idは、マークが置かれているところのPlayItemを指定する8ビットのフィールドである。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図25参照）。

【0190】character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示した値に対応する。name_lengthの8ビットフィ

ールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。Mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どのような値が設定されても良い。

【0191】ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される（後述）。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない事を示す。

【0192】次に、Clip information fileについて説明する。zzzzz.clpi（Clip information fileファイル）は、図45に示すように6個のオブジェクトから構成される。それらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()、およびMakersPrivateData()である。AVストリーム（Clip AVストリームまたはBridge-Clip AV stream）とそれに対応するClip Informationファイルは、同じ数字列の“zzzzz”が使用される。

【0193】図45に示したzzzzz.clpi（Clip information fileファイル）のシンタクスについて説明するに、ClipInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0194】STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0195】ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。MakersPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word（パディングワード）は、zzzzz.clpiファイルのシンタクスにしたがって挿入される。N1,

N2, N3, N4、およびN5は、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしても良い。

【0196】次に、ClipInfoについて説明する。図46は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル（Clip AVストリームまたはBridge-Clip AVストリームファイル）の属性情報をストアする。

【0197】図46に示したClipInfoのシンタクスについて説明するに、version_numberは、このClipInfo()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す32ビットの符号なし整数である。Clip_stream_typeの8ビットのフィールドは、図47に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

【0198】offset_SPNの32ビットのフィールドは、AVストリーム（Clip AVストリームまたはBridge-Clip AVストリーム）ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録される時、このoffset_SPNは0でなければならない。

【0199】図48に示すように、AVストリームファイルのはじめの部分が編集によって消去された時、offset_SPNは、ゼロ以外の値をとっても良い。本実施の形態では、offset_SPNを参照する相対ソースパケット番号（相対アドレス）が、しばしば、RSPN_xxx（xxxは変形する。例。RSPN_EP_start）の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

【0200】AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数（SPN_xxx）は、次式で算出される。

$$SPN_xxx = RSPN_xxx - offset_SPN$$

図48に、offset_SPNが4である場合の例を示す。

【0201】TS_recording_rateは、24ビットの符号なし整数であり、この値は、DVR ドライブ（書き込み部22）へまたはDVR ドライブ（読み出し部28）からのAVストリームの必要な入出力のビットレートを与える。record_time_and_dateは、Clipに対応するAVストリームが記録された時の日時をストアする56ビットのフィールドであり、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、2001/12/23:01:02:03は、“

0x20011223010203”と符号化される。

【0202】durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間／分／秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、01:45:30は、“0x014530”と符号化される。

【0203】time_controlled_flagのフラグは、AVストリームファイルの記録モードを示す。このtime_controlled_flagが1である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$TS_average_rate * 192 / 188 * (t - start_time) - \alpha <= size_clip(t)$
 $<= TS_average_rate * 192 / 188 * (t - start_time) + \alpha$
 ここで、TS_average_rateは、AVストリームファイルのトランスポートストリームの平均ビットレートをbytes/secondの単位で表したものである。

【0204】また、上式において、tは、秒単位で表される時間を示し、start_timeは、AVストリームファイルの最初のソースパケットが記録された時の時刻であり、秒単位で表される。size_clip(t)は、時刻tにおけるAVストリームファイルのサイズをバイト単位で表したものであり、例えば、start_timeから時刻tまでに10個のソースパケットが記録された場合、size_clip(t)は $10 * 192 / 188$ バイトである。 α は、TS_average_rateに依存する定数である。

【0205】time_controlled_flagが0にセットされている場合、記録モードは、記録の時間経過とAVストリームのファイルサイズが比例するように制御していないことを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

【0206】TS_average_rateは、time_controlled_flagが1にセットされている場合、この24ビットのフィールドは、上式で用いているTS_average_rateの値を示す。time_controlled_flagが0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。まずトランスポートレートをTS_recording_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しない事によって、間欠的にトランスポートパケットを符号化する。

【0207】RSPN_arrival_time_discontinuityの32ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、B

ridge-Clip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した
 $SPN_xxx = RSPN_xxx - offset_SPN$
 に基づいて算出される。

【0208】reserved_for_system_useの144ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが1である時、format_identifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが1である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが1である場合、transport_stream_IDのフィールドが有効であることを示す。is_servce_ID_validのフラグが1である場合、servce_IDのフィールドが有効であることを示す。

【0209】is_country_code_validのフラグが1である時、country_codeのフィールドが有効であることを示す。format_identifierの32ビットフィールドは、トランスポートストリームの中でregistration deascriotor (ISO/IEC13818-1で定義されている) が持つformat_identifierの値を示す。original_network_IDの16ビットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの16ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

【0210】servce_IDの16ビットフィールドは、トランスポートストリームの中で定義されているservce_IDの値を示す。country_codeの24ビットのフィールドは、ISO3166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO8859-1で符号化される。例えば、日本は”JPN”と表され、“0x4A 0x500x4E”と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値’0xFF’がセットされる。

【0211】format_identifier、original_network_ID、transport_stream_ID、servce_ID、country_code、およびstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI(サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能

である。

【0212】次に、STC_Infoについて説明する。ここでは、MPEG-2トランSPORTストリームの中でSTCの不連続点（システムタイムベースの不連続点）を含まない時間区間をSTC_sequenceと称し、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図50は、連続なSTC区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない（ただし、後述するように、Clipの最大時間長は制限されている）。従って、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC_sequenceに分割される。

【0213】STC_Infoは、STCの不連続（システムタイムベースの不連続）が発生する場所のアドレスをストアする。図51を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除くk番目(k>=0)のSTC_sequenceは、k番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、(k+1)番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

【0214】図52は、STC_Infoのシンタクスを示す図である。図52に示したSTC_Infoのシンタクスについて説明するに、version_numberは、このSTC_Info()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0215】lengthは、このlengthフィールドの直後からSTC_Info()の最後までのSTC_Info()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map_typeを示す場合、このlengthフィールドはゼロをセットしても良い。CPI()のCPI_typeがEP_map_typeを示す場合、num_of_STC_sequencesは1以上の値でなければならない。

【0216】num_of_STC_sequencesの8ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-loopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、0から開始される。

【0217】RSPN_STC_startの32ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中

で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしても良い。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述したSPN_xxx = RSPN_xxx - offset_SPNにより算出される。

【0218】次に、図45に示したzzzzz.clipのシンタクス内のProgramInfoについて説明する。図53を参照しながら説明するに、ここでは、Clipの中で次の特徴をもつ時間区間をprogram_sequenceと呼ぶ。まず、PCR_PIDの値が変わらない。次に、ビデオエレメンタリーストリームの数が変化しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。さらに、オーディオエレメンタリーストリームの数が変化しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

【0219】program_sequenceは、同一の時刻において、ただ1つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ1つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

【0220】図54は、ProgramInfoのシンタクスを示す図である。図54に示したProgramInfoのシンタクスを説明するに、version_numberは、このProgramInfo()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0221】lengthは、このlengthフィールドの直後からProgramInfo()の最後までのProgramInfo()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map_typeを示す場合、このlengthフィールドはゼロにセットされても良い。CPI()のCPI_typeがEP_map_typeを示す場合、number_of_programsは1以上の値でなければならない。

【0222】number_of_program_sequencesの8ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが変化しない場合、number_of_program_sequencesは1をセットされなければならない。RSPN_program_sequence_startの32ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

【0223】RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリーム

ファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

【0224】PCR_PIDの16ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの8ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number_of_audiosの8ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【0225】audio_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【0226】なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

【0227】図55は、図54に示したProgramInfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図55に示したVideoCodingInfoのシンタクスを説明するに、video_formatの8ビットフィールドは、図56に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

【0228】frame_rateの8ビットフィールドは、図57に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの8ビットフィールドは、図58に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

【0229】図59は、図54に示したProgramInfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図59に示したAudioCodingInfoのシンタクスを説明するに、audio_codingの8ビットフィールドは、

図60に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

【0230】audio_component_typeの8ビットフィールドは、図61に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

【0231】次に、図45に示したzzzzz.clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには2つのタイプがあり、それらはEP_mapとTU_mapである。図63に示すように、CPI()の中のCPI_typeがEP_map_typeの場合、そのCPI()はEP_mapを含む。図64に示すように、CPI()の中のCPI_typeがTU_map_typeの場合、そのCPI()はTU_mapを含む。1つのAVストリームは、1つのEP_mapまたは一つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならない。

【0232】図65は、CPIのシンタクスを示す図である。図65に示したCPIのシンタクスを説明するに、version_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までのCPI()のバイト数を示す32ビットの符号なし整数である。CPI_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタイプを表す。

【0233】次に、図65に示したCPIのシンタクス内のEP_mapについて説明する。EP_mapには、2つのタイプがあり、それはビデオストリーム用のEP_mapとオーディオストリーム用のEP_mapである。EP_mapの中のEP_map_typeが、EP_mapのタイプを区別する。Clipが1つ以上のビデオストリームを含む場合、ビデオストリーム用のEP_mapが使用されなければならない。Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP_mapが使用されなければならない。

【0234】ビデオストリーム用のEP_mapについて図67を参照して説明する。ビデオストリーム用のEP_mapは、stream_PID、PTS_EP_start、および、RSPN_EP_startというデータを持つ。stream_PIDは、ビデオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、ビデオストリームのシーケンスヘッダから始まるアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

【0235】EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランSPORTパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【0236】オーディオストリーム用のEP_mapは、stream_PID、PTS_EP_start、およびRSPN_EP_startというデータを持つ。stream_PIDは、オーディオストリームを伝送するトランSPORTパケットのPIDを示す。PTS_EP_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startで参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

【0237】EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランSPORTパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【0238】EP_mapとSTC_Infoの関係を説明するに、1つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較する事により、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる（図6.8を参照）。・EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP_map_for_one_stream_PIDを持たねばならない。図6.9に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

【0239】図7.0は、EP_mapのシンタクスを示す図である。図7.0に示したEP_mapのシンタクスを説明するに、EP_typeは、4ビットのフィールドであり、図7.1に示すように、EP_mapのエントリーポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP_typeは0('video')にセットされなければならない。または、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP_typeは1('audio')にセットされなければならない。

【0240】number_of_stream_PIDsの16ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数にもつfor-loopのループ回数を示す。stream_PID(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリーストリーム（ビデオまたはオーディオストリーム）を伝送するトランSPORTパケットのPIDを

示す。EP_typeが0 ('video')に等しい場合、そのエレメンタリーストリームはビデオストリームでなければならない。また、EP_typeが1('audio')に等しい場合、そのエレメンタリーストリームはオーディオストリームでなければならない。

【0241】num_EP_entries(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_Start_address(k): この32ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第1バイト目からの大きさで示される。

【0242】padding_wordは、EP_map()のシンタクスにしたがって挿入されなければならない。XとYは、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取っても良い。

【0243】図7.2は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図7.2に示したEP_map_for_one_stream_PIDのシンタクスを説明するに、PTS_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。EP_typeが1 ('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。

【0244】RSPN_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースパケットの相対アドレスを示す。または、EP_typeが1 ('audio')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第一バイト目を含むソースパケットの相対アドレスを示す。

【0245】RSPN_EP_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN
により算出される。シンタクスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

【0246】次に、TU_mapについて、図7.3を参照して説明する。TU_mapは、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つ

の時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、offset_timeから一定の単位に分割される。その単位を、time_unitと称する。

【0247】AVストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startと称する。TU_map_time_axis上において、k (k>=0)番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

$$TU_{start_time}(k) = offset_time + k * time_unit_size$$

TU_start_time(k)は、45kHzの精度を持つ。

【0248】図74は、TU_mapのシンタクスを示す図である。図74に示したTU_mapのシンタクスを説明するに、offset_timeの32bit長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clip中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset_timeはゼロにセットされなければならない。

【0249】time_unit_sizeの32ビットフィールドは、time_unitの大きさを与えるものであり、それは27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。time_unit_sizeは、1秒以下 (time_unit_size<=45000) にすることが良い。number_of_time_unit_entriesの32ビットフィールドは、TU_map()の中にストアされているtime_unitのエントリー数を示す。

【0250】RSPN_time_unit_startの32ビットフィールドは、AVストリームの中でそれぞれのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

$$SPN_{xxx} = RSPN_{xxx} - offset_SPN$$

により算出される。シンタクスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k+1)番目のtime_unitの中にソースパケットが何もない場合、(k+1)番目のRSPN_time_unit_startは、k番目のRSPN_time_unit_startと等しくなければならない。

【0251】図45に示したzzzz.clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置1）によって

セットされるものであり、ユーザによってセットされるものではない。

【0252】図75は、ClipMarkのシンタクスを示す図である。図75に示したClipMarkのシンタクスを説明するに、version_numberは、このClipMark()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0253】lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す32ビットの符号なし整数である。number_of_ClipMarksは、ClipMarkの中にストアされているマークの個数を示す16ビットの符号なし整数。number_of_ClipMarksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図76に示すテーブルに従って符号化される。

【0254】mark_time_stampは、32ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図77に示すように、Playlist()の中のCPI_typeにより異なる。

【0255】STC_sequence_idは、CPI()の中のCPI_typeがEP_map_typeを示す場合、この8ビットのフィールドは、mark_time_stampが置かれているところのSTC連続区間のSTC_sequence_idを示す。CPI()の中のCPI_typeがTU_map_typeを示す場合、この8ビットのフィールドは何も意味を持たず、ゼロにセットされる。character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

【0256】name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0257】ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

【0258】図78は、図75に代わるClipMarkの他のシンタクスを示す図であり、図79は、その場合における、図76に代わるmark_typeのテーブルの例を示す。r

`eserved_for_maker_ID`は、`mark_type`が、0xC0から0xFFの値を示す時に、その`mark_type`を定義しているメーカーのメーカーIDを示す16ビットのフィールドである。メーカーIDは、DVRフォーマットライセンサーが指定する。`mark_entry()`は、マーク点に指定されたポイントを示す情報であり、そのシンタクスの詳細は後述する。`representative_picture_entry()`は、`mark_entry()`によって示されるマークを代表する画像のポイントを示す情報であり、そのシンタクスの詳細は後述する。

【0259】`ClipMark`は、ユーザーがAVストリームを再生するときに、その内容を視覚的に検索できるようにするために用いられる。DVRプレーヤは、GUI(グラフィカルユーザーインターフェース)を使用して、`ClipMark`の情報をユーザーに提示する。`ClipMark`の情報を視覚的に表示するためには、`mark_entry()`が示すピクチャよりもむしろ`representative_picture_entry()`が示すピクチャを示したほうが良い。

【0260】図80に、`mark_entry()`と`representative_picture_entry()`の例を示す。例えば、あるプログラムが開始してから、しばらくした後(数秒後)、そのプログラムの番組名(タイトル)が表示されるとする。`ClipMark`を作るときは、`mark_entry()`は、そのプログラムの開始ポイントに置き、`representative_picture_entry()`は、そのプログラムの番組名(タイトル)が表示されるポイントに置くようにしても良い。

【0261】DVRプレーヤは、`representative_picture_entry`の画像をGUIに表示し、ユーザーがその画像を指定すると、DVRプレーヤは、`mark_entry`の置かれたポイントから再生を開始する。

【0262】`mark_entry()`および`representative_picture_entry()`のシンタクスを、図81に示す。

【0263】`mark_time_stamp`は、32ビットフィールドであり、`mark_entry()`の場合はマークが指定されたポイントを示すタイムスタンプをストアし、また`representative_picture_entry()`の場合、`mark_entry()`によって示されるマークを代表する画像のポイントを示すタイムスタンプをストアする。

【0264】次に、`ClipMark`を指定するために、PTSによるタイムスタンプベースの情報を使用するのではなく、アドレスベースの情報を使用する場合の`mark_entry()`と`representative_picture_entry()`のシンタクスの例を図82に示す。

【0265】`RSPN_ref_EP_start`は、`mark_entry()`の場合、AVストリームの中でマーク点のピクチャをデコードするためのストリームのエントリーポイントを示すソースパケットの相対アドレスを示す。また、`representative_picture_entry()`の場合、`mark_entry()`によって示されるマークを代表するピクチャをデコードするためのストリームのエントリーポイントを示すソースパケットの相対アドレスを示す。`RSPN_ref_EP_start`の値は、`EP_ma`

`p`の中に`RSPN_EP_start`としてストアされていなければならず、かつ、その`RSPN_EP_start`に対応する`PTS_EP_start`の値は、`EP_map`の中で、マーク点のピクチャの`PTS`より過去で最も近い値でなければならない。

【0266】`offset_num_pictures`は、32ビットのフィールドであり、`RSPN_ref_EP_start`により参照されるピクチャから表示順序でマーク点で示されるピクチャまでのオフセットのピクチャ数を示す。この数は、ゼロからカウントされる。図83の例の場合、`offset_num_pictures`は6となる。

【0267】次に、`ClipMark`を指定するために、アドレスベースの情報を使用する場合の`mark_entry()`と`representative_picture_entry()`のシンタクスの別の例を図84に示す。

【0268】`RSPN_mark_point`は、`mark_entry()`の場合、AVストリームの中で、そのマークが参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。また、`representative_picture_entry()`の場合、`mark_entry()`によって示されるマークを代表する符号化ピクチャの第1バイト目を含むソースパケットの相対アドレスを示す。

【0269】`RSPN_mark_point`は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットから`Clip Information file`において定義される`offset_SPN`の値を初期値としてカウントされる。

【0270】図85を用いて、`ClipMark`と`EP_map`の関係を説明する。この例の場合、`EP_map`が、エントリーポイントのアドレスとして`I0, I1, In`を指定しており、これらのアドレスからシーケンスヘッダに続く1ピクチャが開始しているとする。`ClipMark`が、あるマークのアドレスとして、`M1`を指定している時、そのソースパケットから開始しているピクチャをデコードできるためには、`M1`のアドレスより前で最も近いエントリーポイントである`I1`からデータを読み出し開始すれば良い。

【0271】`MakersPrivateData`については、図22を参照して既に説明したので、その説明は省略する。

【0272】次に、サムネイルインフォメーション(`Thumbnail Information`)について説明する。サムネイル画像は、`menu.thmb`ファイルまたは`mark.thmb`ファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ1つの`Thumbnail()`を持つ。`menu.thmb`ファイルは、メニュー・サムネイル画像、すなわち`Volume`を代表する画像、および、それぞれの`PlayList`を代表する画像をストアする。すべてのメニュー・サムネイルは、ただ1つの`menu.thmb`ファイルにストアされる。

【0273】`mark.thmb`ファイルは、マーク・サムネイル画像、すなわちマーク点を表すピクチャをストアする。すべての`PlayList`および`Clip`に対するすべてのマーク・サムネイルは、ただ1つの`mark.thmb`ファイルにストアさ

れる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は一つのtn_blockに格納される。1つの画像データは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。1つのサムネイル画像のバイト長は可変である。

【0274】図86は、menu.thmbとmark.thmbのシンタクスを示す図であり、図87は、図86に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図87に示したThumbnailのシンタクスについて説明するに、version_numberは、このThumbnail()のバージョンナンバーを示す4個のキャラクタ一文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0275】lengthは、このlengthフィールドの直後からThumbnail()の最後までのMakersPrivateData()のバイト数を示す32ビットの符号なし整数である。tn_blocks_start_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。相対バイト数はゼロからカウントされる。number_of_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリー数を与える16ビットの符号なし整数である。

【0276】tn_block_sizeは、1024バイトを単位として、1つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size=1ならば、それは1つのtn_blockの大きさが1024バイトであることを示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエントリ数を表す116ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデックス番号を表す16ビットの符号なし整数である。thumbnail_indexとして、0xFFFFという値を使用してはならない。thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlayList()、PlayListMark()、およびClipMark()の中のref_thumbnail_indexによって参照される。

【0277】thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図88に示すような値をとる。表中のDCFとPNGは“menu.thmb”内でのみ許される。マークサムネイルは、値“0x00”(MPEG-2 Video 1-picture)をとらなければならない。

【0278】picture_data_sizeは、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_blockのtn_block番号を表す16ビッ

トの符号なし整数である。サムネイル画像データの先頭は、tb_blockの先頭と一致していかなければならない。tn_block番号は、0から始まり、tn_blockのforループ中の変数kの値に関係する。

【0279】x_picture_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。y_picture_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。tn_blockは、サムネイル画像がストアされる領域である。Thumbnail()の中のすべてのtn_blockは、同じサイズ(固定長)であり、その大きさはtn_block_sizeによって定義される。

【0280】図89は、サムネイル画像データがどのようにtn_blockに格納されるかを模式的に表した図である。図89のように、各サムネイル画像データはtn_blockの先頭から始まり、1tn_blockを超える大きさの場合は、連続する次のtn_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡単な処理により対応する事ができるようになる。

【0281】次に、AVストリームファイルについて説明する。AVストリームファイルは、“M2TS”ディレクトリ(図14)にストアされる。AVストリームファイルには、2つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

【0282】まず、DVR MPEG-2 トランスポートストリームについて説明する。DVR MPEG-2トランスポートストリームの構造は、図90に示すようになっている。AVストリームファイルは、DVR MPEG2トランスポートストリームの構造を持つ。DVR MPEG2トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144バイト(2048*3バイト)である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、192バイト長である。一つのソースパケットは、TP_extra_headerとトランスポートパケットから成る。TP_extra_headerは、4バイト長であり、またトランスポートパケットは、188バイト長である。

【0283】1つのAligned unitは、32個のソースパケットから成る。DVR MPEG2トランスポートストリームの中の最後のAligned unitも、また32個のソースパケットから成る。よって、DVR MPEG2トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が32の倍数でない時、ヌルパケット(PID=0xFFFFのトランスポートパケット)を持ったソ

スパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG2トランSPORTストリームに余分な情報を付加してはならない。

【0284】図91に、DVR MPEG-2トランSPORTストリームのレコーダモデルを示す。図91に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランSPORTストリームは、このモデルに従う。

【0285】MPEG-2トランSPORTストリームの入力タイミングについて説明する。入力MPEG2トランSPORTストリームは、フルトランSPORTストリームまたはパーシャルトランSPORTストリームである。入力されるMPEG2トランSPORTストリームは、ISO/IEC13818-1またはISO/IEC13818-9に従っていなければならない。MPEG2トランSPORTストリームのi番目のバイトは、T-STD (ISO/IEC 13818-1で規定されるTransport stream system target decoder) 51とソースパケットライザ (source packetizer) 54へ、時刻t(i)に同時に投入される。Rp_kは、トランSPORTパケットの投入レートの瞬時的な最大値である。

【0286】27MHz PLL52は、27MHzクロックの周波数を発生する。27MHzクロックの周波数は、MPEG-2トランSPORTストリームのPCR (Program Clock Reference)の値にロックされる。アライバルタイムクロックカウンタ (arrival time clock counter) 53は、27MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival_time_clock(i)は、時刻t(i)におけるarrival time clockcounter 53のカウント値である。

【0287】source packetizer 54は、すべてのトランSPORTパケットにTP_extra_headerを付加し、ソースパケットを作る。Arrival_time_stampは、トランSPORTパケットの第1バイト目がT-STD 51とソースパケットライザ 54の両方へ到着する時刻を表す。Arrival_time_stamp(k)は、次式で示されるようにArrival_time_clock(k)のサンプル値であり、ここで、kはトランSPORTパケットの第1バイト目を示す。

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

【0288】2つの連続して投入されるトランSPORTパケットの時間間隔が、230/27000000秒 (約40秒) 以上になる場合、その2つのトランSPORTパケットのarrival_time_stampの差分は、230/27000000秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

【0289】スムージングバッファ (smoothing buffer) 55は、入力トランSPORTストリームのビットレートをスムージングする。スムージングバッファ 55は、オーバーフローしてはならない。Rmaxは、スムージングバッファ 55が空でない時のスムージングバッファ 55からのソースパケットの出力ビットレートである。

スムージングバッファ 55が空である時、スムージングバッファ 55からの出力ビットレートはゼロである。

【0290】次に、DVR MPEG-2トランSPORTストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

$$R_{\max} = TS_recording_rate * 192/188$$

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

【0291】入力トランSPORTストリームがSESFトランSPORTストリームの場合、Rp_kは、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateに等しくなければならない。入力トランSPORTストリームがSESFトランSPORTストリームでない場合、この値はMPEG-2 transport streamのデスクリプター、例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptorなど、において定義される値を参照しても良い。

【0292】入力トランSPORTストリームがSESFトランSPORTストリームの場合、スムージングバッファ 55の大きさ (smoothing buffer size) はゼロである。入力トランSPORTストリームがSESFトランSPORTストリームでない場合、スムージングバッファ 55の大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptorなどにおいて定義される値を参照しても良い。

【0293】記録機 (レコーダ) および記録再生装置1 (プレーヤ) は、十分なサイズのバッファを用意しなければならない。デフォルトのバッファサイズは、1536 bytes である。

【0294】次に、DVR MPEG-2トランSPORTストリームのプレーヤモデルについて説明する。図92は、DVR MPEG-2トランSPORTストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランSPORTストリームは、このモデルに従う。

【0295】27MHz X-tal (クリスタル発振器) 61は、27MHzの周波数を発生する。27MHz周波数の誤差範囲は、+/-30 ppm (27000000 +/- 810 Hz) でなければならない。arrival time clock counter 62は、27MHzの周波数のパルスをカウントするバイナリーカウンターである。arrival_time_clock(i)は、時刻t(i)におけるarrival_time_clock_counter 62のカウント値である。

【0296】smoothing buffer 64において、Rmaxは、スムージングバッファ 64がフルでない時のスムージングバッファ 64へのソースパケットの投入ビットレートである。スムージングバッファ 64がフルである時、ス

ムージングバッファ64への入力ビットレートはゼロである。

【0297】MPEG-2トランSPORTストリームの出力タイミングを説明するに、現在のソースパケットのarrival_time_stampがarrival_time_clock(i)のLSB 30ビットの値と等しい時、そのソースパケットのトランSPORTパケットは、スムージングバッファ64から引き抜かれる。Rp1は、トランSPORTパケットレートの瞬時的な最大値である。スムージングバッファ64は、アンダーフローしてはならない。

【0298】DVR MPEG-2トランSPORTストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランSPORTストリームのレコーダモデルのパラメータと同一である。

【0299】図93は、Source packetのシンタクスを示す図である。transport_packet()は、ISO/IEC 13818-1で規定されるMPEG-2トランSPORTパケットである。図93に示したSource packetのシンタクス内のTP_Extra_headerのシンタクスを図94に示す。図94に示したTP_Extra_headerのシンタクスについて説明するに、copy_permission_indicatorは、トランSPORTパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free、no more copy、copy once、またはcopy prohibitedとすることができます。図95は、copy_permission_indicatorの値と、それによって指定されるモードの関係を示す。

【0300】copy_permission_indicatorは、すべてのトランSPORTパケットに附加される。IEEE1394デジタルインターフェースを使用して入力トランSPORTストリームを記録する場合、copy_permission_indicatorの値は、IEEE1394 isochronouspacket headerの中のEMI(Encryption Mode Indicator)の値に関連付けても良い。IEEE1394デジタルインターフェースを使用しないで入力トランSPORTストリームを記録する場合、copy_permission_indicatorの値は、トランSPORTパケットの中に埋め込まれたCCIの値に関連付けても良い。アナログ信号入力をセルフエンコードする場合、copy_permission_indicatorの値は、アナログ信号のCGMS-Aの値に関連付けても良い。

【0301】arrival_time_stampは、次式

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 2^{30}$$

において、arrival_time_stampによって指定される値を持つ整数値である。

【0302】Clip AVストリームの定義をすると、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランSPORTストリームの構造を持たねばならない。arrival_time_clock(i)は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース(STCベース)の不連続

点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しなければならない。

【0303】Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、26時間でなければならない。この制限は、MPEG2トランSPORTストリームの中にシステムタイムベース(STCベース)の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS(Presentation Time Stamp)が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間)と規定している。

【0304】Bridge-Clip AVストリームの定義をすると、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランSPORTストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランSPORTストリームは、後述する符号化の制限に従わなければならず、かつ後述するDVR-STDに従わなければならない。

【0305】本実施の形態においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ/レコーダに"データの連続供給"と"シームレスな復号処理"を保証する。"データの連続供給"とは、ファイルシステムが、デコーダにバッファのアンダーフローを起こさせる事のないように必要なビットレートでデータを供給する事を保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるよう、データが十分な大きさの連続したブロック単位でストアされるようとする。

【0306】"シームレスな復号処理"とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こせる事なく、ディスクに記録されたオーディオビデオデータを表示できることである。

【0307】シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるよう保証されているかどうかは、現在のPlayItemにおいて定義されているconnection_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

【0308】図96は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図96においては、プレーヤが読み出すストリームデータが、影をつけて示されている。図96に示したTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータか

ら成る。

【0309】TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図96においてIN_time1で図示されている)に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

【0310】また、図96におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータから成る。TS2に含まれるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照されるソースパケットから、現在のPlayItemのOUT_time (図96においてOUT_time2で図示されている)に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【0311】図97は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影をつけて示されている。図97におけるTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータから成る。TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図97においてIN_time1で図示されている)に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。また、図97におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータから成る。

【0312】TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT_time (図97においてOUT_time2で図示されている)に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【0313】図96と図97において、TS1とT2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構

造の制限として、まず、TS1とTS2の中に含まれるプログラムの数は、1でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、1でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1および/またはTS2の中に、上記以外のエレメンタリーストリームまたはプライベートストリームが含まれていても良い。

【0314】ビデオビットストリームの制限について説明する。図98は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT_time1 (Clip1のOUT_time) の後とIN_time2 (Clip2のIN_time) の前に表示される不必要的ピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

【0315】図98に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図99に示す。RSPN_arrival_time_discontinuityより前のBridge-Clipのビデオストリームは、図98のClip1のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成る。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、1つの連続でMPEG 2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【0316】同様にして、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図98のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成る。そして、そのビデオストリームは、正しくデコード開始する事ができ、これに続くClip2のビデオストリームに接続され、1つの連続でMPEG 2規格に従ったエレメンタリーストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならず、それ以外のピクチャはオリジナルのClipからコピーすることができる。

【0317】図98に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図100に示す。Clip1のビデオストリームは、図98のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成り、それは、1つの連続でMPEG 2規格に従ったエレメンタリーストリームとなるように再エンコードされている。同様にして、Clip2のビデオストリームは、図98のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成り、それは、一つの連続でMPEG 2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【0318】ビデオストリームの符号化制限について説明するに、まず、TS1とTS2のビデオストリームのフレー

ムレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GO P Header、そしてI-ピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

【0319】ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレームまたはフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレームまたはフィールドのギャップがあつてはならない。接続点において、トップ一ボトムのフィールドシーケンスは連続でなければならない。3-2フルダウントを使用するエンコードの場合は、"top_field_first" および "repeat_first_field" フラグを書き換える必要があるかもしれない、またはフィールドギャップの発生を防ぐために局所的に再エンコードするようにしても良い。

【0320】オーディオビットストリームの符号化制限について説明するに、TS1とTS2のオーディオのサンプリング周波数は、同じでなければならない。TS1とTS2のオーディオの符号化方法（例、MPEG1レイヤ2、AC-3、SESF LPCM、AAC）は、同じでなければならない。

【0321】次に、MPEG-2トランスポートストリームの符号化制限について説明するに、TS1のオーディオストリームの最後のオーディオフレームは、TS1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2のオーディオストリームの最初のオーディオフレームは、TS2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

【0322】接続点において、オーディオプレゼンテーションユニットのシーケンスにギャップがあつてはならない。図101に示すように、2オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあつても良い。TS2のエレメンタリーストリームを伝送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

【0323】ClipおよびBridge-Clipの制限について説明するに、TS1とTS2は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

【0324】以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ1つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

【0325】BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。

【0326】PlayItemの制限について説明するに、先行するPlayItemのOUT_time（図96、図97において示されるOUT_time1）は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time（図96、図97において示されるIN_time2）は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

【0327】Bridge-Clipを使用する場合のデータアロケーションの制限について、図102を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1（ClipAVストリームファイル）とClip2（Clip AVストリームファイル）に接続されるBridge-Clip AVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

【0328】RSPN_exit_from_previous_Clip以前のClip1（Clip AVストリームファイル）のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip AVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip2（Clip AVストリームファイル）のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

【0329】Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図103を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1（Clip AVストリームファイル）の最後の部分とClip2（Clip AVストリームファイル）の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

【0330】Clip1（Clip AVストリームファイル）の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されなければならない。Clip2（Clip AVストリームファイル）の最初のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。

ならない。

【0331】次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG2トランSPORTストリームの生成および検証の際ににおけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるAVストリームの生成および検証の際ににおけるデコード処理をモデル化するための概念モデルである。

【0332】DVR-STDモデルを図104に示す。図104に示したモデルには、DVR MPEG-2トランSPORTストリームプレーヤモデルが構成要素として含まれている。n, TBn, MBn, EBn, TBsys, Bsys, Rxn, Rbxn, Rxsys, Dn, Dsys, OnおよびPn(k)の表記方法は、ISO/IEC13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリーストリームのインデクス番号である。TBnは、エレメンタリーストリームnのトランSPORTバッファである。

【0333】MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBsysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

【0334】Rxsysは、データがTBsysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシステム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

【0335】DVR-STDのデコーディングプロセスについて説明する。単一のDVR MPEG-2トランSPORTストリームを再生している間は、トランSPORTパケットをTB1, TBnまたはTBsysのバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB1, MB1, EB1, TBn, Bn, TBsysおよびBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

【0336】シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図96に示

した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

【0337】図105は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る時のトランSPORTパケットの入力、復号、表示のタイミングチャートを示す。所定のAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る間には、TS2のアライバルタイムベースの時間軸（図105においてATC2で示される）は、TS1のアライバルタイムベースの時間軸（図105においてATC1で示される）と同じでない。

【0338】また、TS2のシステムタイムベースの時間軸（図105においてSTC2で示される）は、TS1のシステムタイムベースの時間軸（図105においてSTC1で示される）と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあつても良い。

【0339】DVR-STDへの入力タイミングについて説明する。時刻T1までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1, TBn またはTBsysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

【0340】TS1の残りのパケットは、TS_recording_rate(TS1)のビットレートでDVR-STDのTBnまたはTBsysのバッファへ入力されなければならない。ここで、TS_recording_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻T2である。従って、時刻T1からT2までの区間では、ソースパケットのarrival_time_stampは無視される。

【0341】N1をTS1の最後のビデオパケットに続くTS1のトランSPORTパケットのバイト数とすると、時刻T1乃至T2までの時間DT1は、N1バイトがTS_recording_rate(TS1)のビットレートで入力終了するために必要な時間であり、次式により算出される。

$$\Delta T1 = T2 - T1 = N1 / TS_recording_rate (TS1)$$

時刻T1乃至T2までの間は、RXnとRxsysの値は共に、TS_recording_rate(TS1)の値に変化する。このルール以外のバッファリング動作は、T-STDと同じである。

【0342】T2の時刻において、arrival_time_clock_counterは、TS2の最初のソースパケットのarrival_time_stampの値にリセットされる。DVR-STDのTB1, TBn またはTBsysのバッファへの入力タイミングは、TS2のソースパケットのarrival_time_stampによって決定される。RXnとRxsysは共に、T-STDにおいて定義されている値に変化する。

【0343】付加的なオーディオバッファリングおよび

システムデータバッファリングについて説明するに、オーディオデコーダとシステムデコーダは、時刻T₁からT₂までの区間の入力データを処理することができるよう、T-STDで定義されるバッファ量に加えて付加的なバッファ量（約1秒分のデータ量）が必要である。

【0344】ビデオのプレゼンテーションタイミングについて説明するに、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC1は、TS1のシステムタイムベースの時間軸（図105ではSTC1と図示されている）とし、STC2は、TS2のシステムタイムベースの時間軸（図97ではSTC2と図示されている。正確には、STC2は、TS2の最初のPCRがT-STDに入力した時刻から開始する。）とする。

【0345】STC1とSTC2の間のオフセットは、次のように決定される。PTS₁_{end}は、TS1の最後のビデオプレゼンテーションユニットに対応するSTC1上のPTSであり、PTS₂_{start}は、TS2の最初のビデオプレゼンテーションユニットに対応するSTC2上のPTSであり、T_{pp}は、TS1の最後のビデオプレゼンテーションユニットの表示期間とすると、2つのシステムタイムベースの間のオフセットSTC_{delta}は、次式により算出される。

$$STC_{\text{delta}} = PTS1_{\text{end}} + T_{\text{pp}} - PTS2_{\text{start}}$$

【0346】オーディオのプレゼンテーションのタイミングについて説明するに、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあつても良く、それは0乃至2オーディオフレーム未満である（図105に図示されている“audio overlap”を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

【0347】DVR-STDのシステムタイムクロックについて説明するに、時刻T₅において、TS1の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻T₂からT₅の間にオーバーラップしていくても良い。この区間では、DVR-STDは、システムタイムクロックを古いタイムベースの値（STC1）と新しいタイムベースの値（STC2）の間で切り替える。STC2の値は、次式により算出される。

$$STC2 = STC1 - STC_{\text{delta}}$$

【0348】バッファリングの連続性について説明する。STC1_{video_end}は、TS1の最後のビデオパケットの最後のバイトがDVR-STDのTB1へ到着する時のシステムタイムベースSTC1上のSTCの値である。STC2_{video_start}は、TS2の最初のビデオパケットの最初のバイトがDVR-STDのTB1へ到着する時のシステムタイムベースSTC2上のSTCの値である。STC2_{video_end}は、STC1_{video_end}の値をシステムタイムベースSTC2上の値に換算した値であ

る。STC2_{video_end}は、次式により算出される。

$$STC2_{\text{video_end}} = STC1_{\text{video_end}} - STC_{\text{delta}}$$

【0349】DVR-STDに従うために、次の2つの条件を満たす事が要求される。まず、TS2の最初のビデオパケットのTB1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$STC2_{\text{video_start}} > STC1_{\text{video_end}} + \Delta T1$$

この不等式が満たされるように、Clip1および、または、Clip2の部分的なストリームを再エンコードおよび、または、再多重化する必要がある場合は、その必要に応じて行われる。

【0350】次に、STC1とSTC2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS1からのビデオパケットの入力とそれに続くTS2からのビデオパケットの入力は、ビデオバッファをオーバーフローおよびアンダーフローさせてはならない。

【0351】このようなシンタクス、データ構造、規則に基づく事により、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

【0352】なお、本実施の形態は、多重化ストリームとしてMPEG2トランスポートストリームを例にして説明しているが、これに限らず、MPEG2プログラムストリームや米国のDirecTVサービス（商標）で使用されているDSSトランスポートストリームについても適用することができる。

【0353】次に、mark_entry()およびrepresentative_picture_entry()のシンタクスが、図81に示されるような構成である場合における、マーク点で示されるシーンの頭出し再生を行う場合の処理について、図106のフローチャートを参照して、説明する。

【0354】最初にステップS1において、記録再生装置1の制御部23は、記録媒体100から、DVRトランスポートストリームファイルのデータベースであるEP_Map（図70）、STC_Info（図52）、Program_Info（図54）、およびClipMark（図78）を読み出す。

【0355】ステップS2において、制御部23は、ClipMark（図78）のrepresentative_picture_entry（図81）、またはref_thumbnail_indexで参照されるピクチャからサムネイルのリストを作成し、ユーザインターフェース入出力としての端子24から出力し、GUIのメニュー画面上に表示させる。この場合、ref_thumbnail_indexが有効な値を持つ場合、representative_picture_entryよりref_thumbnail_indexが優先される。

【0356】ステップS3において、ユーザが再生開始点のマーク点を指定する。これは、例えば、GUIとして表示されたメニュー画面上の中からユーザがサムネイル

画像を選択することで行われる。制御部23は、この選択操作に対応して、指定されたサムネイルに対応づけられているマーク点を取得する。

【0357】ステップS4において、制御部23は、ステップS3で指定されたmark_entry(図81)のmark_time_stampのPTSと、STC_sequence_idを取得する。

【0358】ステップS5において、制御部23は、STC_Info(図52)から、ステップS4で取得したSTC_sequence_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

【0359】ステップS6において、制御部23は、ステップS5で取得したSTC時間軸が開始するパケット番号と、ステップS4で取得したマーク点のPTSから、マーク点のPTSより時間的に前で、かつ、最も近いエントリーポイント(1ピクチャ)のあるソースパケット番号を取得する。

【0360】ステップS7において、制御部23は、ステップS6で取得したエントリーポイントのあるソースパケット番号から、トランSPORTストリームのデータを読み出し、AVデコーダ27に供給させる。

【0361】ステップS8において、制御部23は、AVデコーダ27を制御し、ステップS4で取得したマーク点のPTSのピクチャから表示を開始させる。

【0362】以上の動作を、図107乃至109を参照してさらに説明する。

【0363】いま、図107に示されているように、DVRトランSPORTストリームファイルは、STC_sequence_id=id0のSTC時間軸を有し、その時間軸が開始するソースパケット番号は、シーン開始点Aのソースパケット番号より小さいものとする。そして、ソースパケット番号BからCまでの間に、CM(コマーシャル)が挿入されているものとする。

【0364】このとき、図70に示されるEP_Mapに対応するEP_Mapには、図108に示されるように、RSPN_EP_startで示されるA, B, Cに対応して、それぞれのPTSが、PTS_EP_startとして、PTS(A), PTS(B), PTS(C)として登録される。

【0365】また、図109に示されるように、図78のClipMarkに対応するClipMarkには、図109に示されるように、シーンスタート、CMスタート、およびCMエンドを表すマークタイプ(図79)0x92, 0x94, 0x95の値に対応して、mark_entryとrepresentative_picture_entryが記録される。

【0366】mark_entryのMark_Time_stampとしては、シーンスタート、CMスタート、およびCMエンドに対応して、それぞれPTS(a1), PTS(b0), PTS(c0)が登録されており、それぞれのSTC_sequence_idは、いずれもid0とされている。

【0367】同様に、Representative_picture_entryのMark_Time_stampとして、シーンスタート、CMスター

ト、およびCMエンドに対応して、それぞれPTS(a2), PTS(b0), PTS(c0)が登録されており、それらはいずれもSTC_sequence_idが、id0とされている。

【0368】PTS(A) < PTS(a1)の場合、ステップS6において、パケット番号Aが取得され、ステップS7において、パケット番号Aから始まるトランSPORTストリームが、AVデコーダ27に供給され、ステップS8において、PTS(a1)のピクチャから表示が開始される。

【0369】次に、図110のフローチャートを参照して、mark_entryとrepresentative_picture_entryのシンタクスが、図81に示されるような構成である場合におけるCMスキップ再生の処理について、図110のフローチャートを参照して説明する。

【0370】ステップS21において、制御部23は、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、およびClipMark(図78)を記録媒体100から読み出す。ステップS22において、ユーザは、ユーザインターフェース入出力としての端子24からCMスキップ再生を指定する。

【0371】ステップS23において、制御部23は、マークタイプ(図79)がCM開始点(0x94)であるマーク情報のPTSと、CM終了点(0x95)であるマーク情報のPTS、並びに対応するSTC_sequence_idを取得する(図81)。

【0372】ステップS24において、制御部23は、STC_Info(図52)からCM開始点と終了点の、STC_sequence_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

【0373】ステップS25において、制御部23は、記録媒体100からトランSPORTストリームを読み出させ、それをAVデコーダ27に供給し、デコードを開始させる。

【0374】ステップS26において、制御部23は、現在の表示画像がCM開始点のPTSの画像か否かを調べる。現在の表示画像がCM開始点のPTSの画像でない場合には、ステップS27に進み、制御部23は、画像の表示が継続される。その後、処理はステップS25に戻り、それ以降の処理が繰り返し実行される。

【0375】ステップS26において、現在の表示画像がCM開始点のPTSの画像であると判定された場合、ステップS28に進み、制御部23は、AVデコーダ27を制御し、デコードおよび表示を停止させる。

【0376】次に、ステップS29において、制御部23は、CM終了点のSTC_sequence_idに対応するSTC時間軸が開始するパケット番号を取得し、そのパケット番号と、ステップS23の処理で取得したCM終了点のPTSとから、その点のPTSより時間的に前で、かつ、最も近いエントリーポイントのあるソースパケット番号を取得する。

【0377】ステップS30において、制御部23は、

ステップS29の処理で取得したエントリーポイントのあるソースパケット番号から、トランSPORTストリームのデータを読み出し、AVデコーダ27に供給させる。

【0378】ステップS31において、制御部23は、AVデコーダ27を制御し、CM終了点のPTSのピクチャから表示を再開させる。

【0379】図107乃至図109を参照して、以上の動作をさらに説明すると、CM開始点とCM終了点は、この例の場合、STC_sequence_id=id0という共通のSTC時間軸上に存在し、そのSTC時間軸が開始するソースパケット番号は、シーンの開始点のソースパケット番号Aより小さいものとされている。

【0380】トランSPORTストリームがデコードされ、ステップS26で、表示時刻がPTS(b0)になったと判定された場合（CM開始点であると判定された場合）、AVデコーダ27により表示が停止される。そして、PTS(C) < PTS(c0)の場合、ステップS30でパケット番号Cのデータから始まるストリームからデコードが再開され、ステップS31において、PTS(c0)のピクチャから表示が再開される。

【0381】なお、この方法は、CMスキップ再生に限らず、一般的にClipMarkで指定される2点間のシーンをスキップして再生する場合にも、適用可能である。

【0382】次に、mark_entryとrepresentative_picture_entryが、図82に示すシンタクス構造である場合における、マーク点で示されるCMの頭出し再生処理について、図112のフローチャートを参照して説明する。

【0383】ステップS41において、制御部23は、EP_map（図70）、STC_Info（図52）、Program_Info（図54）、およびClipMark（図78）の情報を取得する。

【0384】次にステップS42において、制御部23は、ステップS41で読み出したClipMark（図78）に含まれるrepresentative_picture_entry（図82）またはref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面上に表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

【0385】ステップS43において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、ステップS42の処理で表示されたメニュー画面上の中から、ユーザがサムネイル画像を選択し、そのサムネイルに対応づけられているマーク点を指定することで行われる。

【0386】ステップS44において、制御部23は、ステップS43の処理で指定されたマーク点のRSPN_ref_EP_startとoffset_num_pictures（図82）を取得する。

【0387】ステップS45において、制御部23は、

ステップS44で取得したRSPN_ref_EP_startに対応するソースパケット番号からトランSPORTストリームのデータを読み出し、AVデコーダ27に供給させる。

【0388】ステップS46において、制御部23は、AVデコーダ27を制御し、RSPN_ref_EP_startで参照されるピクチャから（表示はしないで）、表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_picturesになったとき、そのピクチャから表示を開始させる。

【0389】以上の処理を、図113乃至図115を参照して、さらに説明する。この例においては、DVRトランSPORTストリームファイルは、ソースパケット番号Aからシーンが開始しており、ソースパケット番号BからソースパケットCまでCMが挿入されている。このため、図114に示されるように、EP_mapには、RSPN_EP_startとしてのA、B、Cに対応して、PTS_EP_startとして、PTS(A)、PTS(B)、PTS(C)が登録されている。

【0390】また、図115に示されるように、シーンスタート、CMスタート、およびCMエンドのマークタイプに対応して、mark_entryとrepresentative_picture_entryが登録されている。mark_entryには、シーンスタート、CMスタート、およびCMエンドに対応して、RSPN_ref_EP_startとして、それぞれA、B、Cが登録され、offset_num_picturesとして、M1、N1、N2が登録されている。同様に、representative_picture_entryには、RSPN_ref_EP_startとして、シーンスタート、CMスタート、およびCMエンドに対応して、それぞれA、B、Cが登録され、offset_num_picturesとして、M2、N1、N2がそれぞれ登録されている。

【0391】シーンスタートに当たるピクチャから頭出して再生が指令された場合、パケット番号Aのデータから始まるストリームからデコードが開始され、PTS(A)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップをしていき、offset_num_picturesが、M1の値になったとき、そのピクチャから表示が開始される。

【0392】さらに、mark_entryとrepresentative_picture_entryのシンタクスが、図82に示される構成である場合におけるCMスキップ再生の処理について、図116のフローチャートを参照して説明する。

【0393】ステップS61において、制御部23は、EP_map（図70）、STC_Info（図52）、Program_Info（図54）、およびClipMark（図78）の情報を取得する。

【0394】ステップS62において、ユーザがCMスキップ再生を指令すると、ステップS63において、制御部23は、マークタイプ（図79）がCM開始点とCM終了点である各点のマーク情報として、RSPN_ref_EP_STARTとoffset_num_pictures（図82）を取得する。そして、CM開始点のデータは、RSPN_ref_EP_start(1), offse

`t_num_pictures(1)`とされ、CM終了点のデータは、`RSPN_ref_EP_start(2)`、`offset_num_pictures(2)`とされる。

【0395】ステップS64において、制御部23は、`RSPN_ref_EP_start(1)`、`RSPN_ref_EP_start(2)`に対応するPTSを`EP_map`（図70）から取得する。

【0396】ステップS65において、制御部23は、トランSPORTストリームを記録媒体100から読み出させ、AVデコーダ27に供給させる。

【0397】ステップS66において、制御部23は、現在の表示画像が`RSPN_ref_EP_start(1)`に対応するPTSのピクチャであるか否かを判定し、現在の表示画像が`RSPN_ref_EP_start(1)`に対応するPTSのピクチャでない場合には、ステップS67に進み、ピクチャをそのまま継続的に表示させる。その後、処理はステップS65に戻り、それ以降の処理が繰り返し実行される。

【0398】ステップS66において、現在の表示画像が`RSPN_ref_EP_start(1)`に対応するPTSのピクチャであると判定された場合、ステップS68に進み、制御部23は、AVデコーダ27を制御し、`RSPN_ref_EP_start(1)`に対応するPTSのピクチャから表示するピクチャをカウントアップしていき、カウント値が`offset_num_pictures(1)`になったとき、表示を停止させる。

【0399】ステップS69において、制御部23は、`RSPN_ref_EP_start(2)`のソースパケット番号からトランSPORTストリームのデータを読み出し、AVデコーダ27に供給させる。

【0400】ステップS70において、制御部23は、AVデコーダ27を制御し、`RSPN_ref_EP_start(2)`に対応するPTSのピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、カウント値が`offset_num_pictures(2)`になったとき、そのピクチャから表示を開始させる。

【0401】以上の動作を、図113乃至図115を参照してさらに説明すると、まず、`EP_map`（図114）をもとに、パケット番号B、Cに対応する時刻PTS(B)、PTS(C)が得られる。そして、Clip AV streamがデコードされていき、表示時刻がPTS(B)になったとき、PTS(B)のピクチャから表示ピクチャがカウントアップされ、その値がN1（図115）になったとき、表示が停止される。

【0402】さらに、パケット番号Cのデータから始まるストリームからデコードが再開され、PTS(C)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、その値がN2（図115）になったとき、そのピクチャから表示が再開される。

【0403】以上の処理は、CMスキップ再生に限らず、ClipMarkで指定された2点間のシーンをスキップさせて再生する場合にも、適用可能である。

【0404】次に、`mark_entry`と`representative_picture_entry`のシンタクスが、図84に示すような構成である場合における、マーク点で示されるシーンの頭出し再

生処理について、図118のフローチャートを参照して説明する。

【0405】ステップS81において、`EP_map`（図70）、`STC_Info`（図52）、`Program_Info`（図54）、並びに`ClipMark`（図78）の情報が取得される。

【0406】ステップS82において、制御部23は、`ClipMark`（図78）の`representative_picture_entry`または`ref_thumbnail_index`で参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面として表示させる。`ref_thumbnail_index`が有効な値を有する場合、`representative_picture_entry`より`ref_thumbnail_index`が優先される。

【0407】ステップS83において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、メニュー画面上の中からユーザがサムネイル画像を選択し、そのサムネイルに対応づけられているマーク点を指定することで行われる。

【0408】ステップS84において、制御部23は、ユーザから指定された`mark_entry`の`RSPN_mark_point`（図84）を取得する。

【0409】ステップS85において、制御部23は、マーク点の`RSPN_mark_point`より前にあり、かつ、最も近いエントリーポイントのソースパケット番号を、`EP_map`（図70）から取得する。

【0410】ステップS86において、制御部23は、ステップS85で取得したエントリーポイントに対応するソースパケット番号からトランSPORTストリームのデータを読み出し、AVデコーダ27に供給させる。

【0411】ステップS87において、制御部23は、AVデコーダ27を制御し、`RSPN_mark_point`で参照されるピクチャから表示を開始させる。

【0412】以上の処理を、図119乃至図121を参照してさらに説明する。この例においては、DVRトランSPORTストリームファイルが、ソースパケットAでシーンスタートし、ソースパケット番号BからCまでCMが挿入されている。このため、図120の`EP_map`には、`RSPN_EP_start`としてのA、B、Cに対応して、`PTS_EP_start`がそれぞれPTS(A)、PTS(B)、PTS(C)として登録されている。また、図121に示されるClipMarkに、シーンスタート、CMスタート、およびCMエンドに対応して、`mark_entry`の`RSPN_mark_point`として、a1, b1, c1が、また、`representative_picture_entry`の`RSPN_mark_point`として、a2, b1, c1が、それぞれ登録されている。

【0413】シーンスタートにあたるピクチャから頭出して再生する場合、パケット番号A < a1とすると、パケット番号Aのデータから始まるストリームからデコードが開始され、ソースパケット番号a1に対応するピクチャから表示が開始される。

【0414】次に、`mark_entry`と`representative_picture_entry`

re_entryのシンタクスが、図84に示されるような構成である場合におけるCMスキップ再生の処理について、図122と図123のフローチャートを参照して説明する。

【0415】ステップS101において、制御部23は、EP_map(図70)、STC_Info(図52)、Program_Info(図54)、並びにClipMark(図70)の情報を取得する。

【0416】ステップS102において、ユーザは、CMスキップ再生を指定する。

【0417】ステップS103において、制御部23は、マークタイプ(図79)がCM開始点とCM終了点である各点のマーク情報のRSPN_mark_point(図84)を取得する。そして、制御部23は、CM開始点のデータをRSPN_mark_point(1)とし、CM終了点のデータをRSPN_mark_point(2)とする。

【0418】ステップS104において、制御部23は、記録媒体100からトランスポートストリームを読み出させ、AVデコーダ27に出力し、デコードさせる。

【0419】ステップS105において、制御部23は、現在の表示画像がRSPN_mark_point(1)に対応するピクチャであるか否かを判定し、現在の表示画像がRSPN_mark_point(1)に対応するピクチャでない場合には、ステップS106に進み、そのままピクチャを継続的に表示させる。その後、処理はステップS104に戻り、それ以降の処理が繰り返し実行される。

【0420】ステップS105において、現在の表示画像がRSPN_mark_point(1)に対応するピクチャであると判定された場合、ステップS107に進み、制御部23はAVデコーダ27を制御し、デコードおよび表示を停止させる。

【0421】次に、ステップS108において、RSPN_mark_point(2)より前にあり、かつ、最も近いエントリーポイントのあるソースパケット番号がEP_map(図70)から取得される。

【0422】ステップS109において、制御部23は、ステップS108で取得したエントリーポイントに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給される。

【0423】ステップS110において、制御部23は、AVデコーダ27を制御し、RSPN_mark_point(2)で参照されるピクチャから表示を再開させる。

【0424】以上の処理を図119乃至図121の例でさらに説明すると、Clip AV streamをデコードして行き、ソースパケット番号b1(図121)に対応する表示ピクチャになったとき、表示が停止される。そして、ソースパケット番号C<ソースパケット番号c1とすると、パケット番号Cのデータから始まるストリームからデコードが再開され、ソースパケット番号c1に対応す

るピクチャになったとき、そのピクチャから表示が再開される。

【0425】以上のようにして、図124に示されるように、PlayList上で、タイムスタンプにより所定の位置を指定し、このタイムスタンプを各ClipのClip Informationにおいて、データアドレスに変換し、Clip AV streamの所定の位置にアクセスすることができる。

【0426】より具体的には、図125に示されるように、PlayList上において、PlayListMarkとしてブックマークやリリューム点を、ユーザが時間軸上のタイムスタンプとして指定すると、そのPlayListは再生するとき、そのPlayListが参照しているClipのClipMarkを使用して、Clip AV streamのシーン開始点やシーン終了点にアクセスすることができる。

【0427】なお、ClipMarkのシンタクスは、図78の例に替えて、図126に示すようにすることもできる。

【0428】この例においては、RSPN_markが、図78のreserved_for_MakerID, mark_entry()、およびresetative_picture_entry()に替えて挿入されている。このRSPN_markの32ビットのフィールドは、AVストリームファイル上で、そのマークが参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。RSPN_markは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義され、offset_SPNの値を初期値としてカウントされる。

【0429】その他の構成は、図78における場合と同様である。

【0430】ClipMarkのシンタクスは、さらに図127に示すように構成することもできる。この例においては、図126におけるRSPN_markの代わりに、RSPN_ref_EP_startとoffset_num_picturesが挿入されている。これらは、図82に示した場合と同様のものである。

【0431】図128は、アナログAV信号をエンコードして記録する場合、図81に示したシンタクスのClip Markの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS200において、解析部14は端子11, 12からの入力AV信号を解析して、特徴点を検出する。特徴点は、AVストリームの内容に起因する特徴的なシーンを指定し、例えば、番組の頭だし点やシーンチェンジ点などである。

【0432】ステップS201において、制御部23は特徴点の画像のPTSを取得する。ステップS202において、制御部23は、特徴点の情報をClipMarkにストアする。具体的には、本実施の形態のClipMarkのシンタクスとセマンティクスで説明した情報をストアする。ステップS203において、Clip Information fileとClip AV stream fileがディスクに記録される。

【0433】図129は、ディジタルインターフェースから入力されたトランSPORTストリームを記録する場合、図81に示したシンタクスのClipMarkの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS211において、デマルチプレクサ26、および、制御部23は、記録するプログラムのエレメンタリストリームPIDを取得する。解析対象のエレメンタリストリームが複数ある場合、全てのエレメンタリストリームPIDが取得される。

【0434】ステップS212で、デマルチプレクサ26は、端子13から入力されるトランSPORTストリームのプログラムからエレメンタリストリームを分離し、それをAVデコーダ27がAV信号にデコードする。ステップS213において、解析部14は、上記AV信号を解析して特徴点を検出する。

【0435】ステップS214において、制御部23は、特徴点の画像のPTSと、それが属するSTCのSTC-sequence-idを取得する。ステップS215で、制御部23は、特徴点の情報をClipMarkにストアする。具体的には、本実施の形態におけるClipMarkのシンタクスとセマンティクスで説明した情報をストアする。

【0436】ステップS216において、Clip Information fileとClip AV stream fileがディスクに記録される。

【0437】図128に示したフローチャート、および、図129に示したフローチャートのようにして、AVストリームファイル、すなわちClip AVストリームファイルの中の特徴的な画像を指し示すマークをストアするClipMarkが、前記AVストリームの管理情報データファイル、すなわちClip Informationファイルに記録される。

【0438】図130は、Real Playlistの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS221において、制御部23はClip AVストリームを記録する。ステップS222において、制御部23は、上記Clipの全ての再生可能範囲をカバーするPlayItemからなるPlaylist()を作成する。Clipの中にSTC不連続点があり、Playlist()が2つ以上のPlayItemからなる場合、PlayItem間のconnection_conditionもまた決定される。

【0439】ステップS223において、制御部23は、UIAppInfoPlaylist()を作成する。ステップS224において、制御部23は、PlaylistMarkを作成する。ステップS225において、制御部23は、MakersPrivateDataを作成する。ステップS226において、制御部23は、Real Playlistファイルを記録する。

【0440】このようにして、新規にClip AVストリームを記録する毎に、1つのReal Playlistファイルが作られる。

【0441】図131は、Virtual Playlistの作成について説明するフローチャートである。ステップS231において、ユーザーインターフェースを通して、ディスクに記録されている1つのReal Playlistの再生が指定される。そして、そのReal Playlistの再生範囲の中から、ユーザーインターフェースを通して、IN点とOUT点で示される再生区間が指定される。

【0442】ステップS232において、制御部23は、ユーザによる再生範囲の指定操作がすべて終了したか否かを判断する。ステップS232において、ユーザによる再生範囲の指定操作はまだ終了していないと判断された場合、ステップS231に戻り、それ以降の処理が繰り返され、終了したと判断された場合、ステップS233に進む。

【0443】ステップS233において、連続して再生される2つの再生区間の間の接続状態(connection_condition)が、ユーザーがユーザーインターフェースを通して決定されるか、または制御部23により決定される。ステップS234において、ユーザーインターフェースを通して、ユーザがサブパス(アフレコ用オーディオ)情報を指定する。ユーザーがサブパスを作成しない場合、ステップS234における処理はスキップされる。

【0444】ステップS235において、制御部23は、ユーザが指定した再生範囲情報、およびconnection_conditionに基づいて、Playlist()を作成する。ステップS236において、制御部23はUIAppInfoPlaylist()を作成する。ステップS237において、制御部23は、PlaylistMarkを作成する。ステップS238において、制御部23は、MakersPrivateDataを作成する。ステップS239において、制御部23は、Virtual Playlistファイルを、ディスクに記録させる。

【0445】このようにして、ディスクに記録されているReal Playlistの再生範囲の中から、ユーザが、見たい再生区間を選択し、その再生区間をグループ化したものの毎に、1つのVirtual Playlistファイルが作成される。

【0446】図132は、Playlistの再生について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS241において、制御部23は、Info.dvr, Clip Information file, Playlist fileおよびサムネールファイルの情報を取得し、ディスクに記録されているPlaylistの一覧を示すGUI画面を作成し、ユーザーインターフェースを通して、GUIに表示する。

【0447】ステップS242において、ユーザーインターフェースを通して、ユーザが1つのPlaylistの再生を制御部23に指示する。ステップS243において、制御部23は、現在のPlayItemのSTC-sequence-idとIN_timeのPTSから、IN_timeより時間的に前で最も近いエントリーポイントのあるソースパケット番号を取得す

る。ステップS244において、制御部23は、上記エントリーポイントのあるソースパケット番号からAVストリームのデータを読み出し、AVデコーダ27へ供給する。

【0448】上記PlayItemの時間的に前にPlayItemの再生があった場合、ステップS245において、制御部23は、そのPlayItemとの表示の接続処理をconnection_conditionに従って行なわれるように制御を行う。ステップS246において、AVデコーダ27は、IN_timeのPTSのピクチャから表示を開始する。

【0449】ステップS247において、AVデコーダ27は、AVストリームのデコードを継続的に行う。ステップS248において、制御部23は、現在表示の画像が、OUT_timeのPTSの画像か否かを判断する。ステップS248において、現在表示の画像は、OUT_timeのPTSの画像であると判断された場合、ステップS250に進み、PTSの画像ではないと判断された場合、ステップS249に進む。

【0450】ステップS249において、PTSの画像であると判断された画像を表示するための処理が実行され、その後ステップS247に戻り、それ以降の処理が繰り返される。一方、ステップS250においては、制御部23により、現在のPlayItemがPlayListの中で最後のPlayItemか否かが判断される。ステップS250において、現在のPlayItemがPlayListの中で最後のPlayItemであると判断された場合、図132に示したフローチャートの処理は終了され、最後のPlayItemではないと判断された場合、ステップS243に戻り、それ以降の処理が繰り返される。

【0451】図133は、PlayListMarkの作成について説明するフローチャートである。図1の記録再生装置1のブロック図を参照しながら説明する。ステップS261において、制御部23は、Info.dvr, Clip Information file, PlayList fileおよびThumbnail fileの情報を取得し、ディスクに記録されているPlayListの一覧を示すGUI画面を作成し、ユーザーインターフェースを通して、GUIに表示する。

【0452】ステップS262において、ユーザーインターフェースを通して、ユーザにより1つのPlayListの再生が制御部23に指示される。ステップS263において、再生部3は、指示されたPlayListの再生を開始する(図132のフローチャートを参照して説明したように行われる)。

【0453】ステップS264において、ユーザーインターフェースを通して、ユーザにより、お気に入りのシンのところにマークのセットが制御部23に指示される。ステップS265において、制御部23は、マークのPTSと、それが属するPlayItemのPlayItem_idを取得する。

【0454】ステップS266において、制御部23

は、マークの情報をPlayListMark()にストアする。ステップS267において、PlayListファイルがディスクに記録される。

【0455】このようにして、PlayListの再生範囲の中からユーザが指定したマーク点、または、そのPlayListを再生するときのResume点を示すマークをストアするPlayListMarkを、PlayListファイルに記録される。

【0456】図134は、PlayListが再生される時、PlayListMarkおよびそのPlayListが参照するClipのClipMarkが使用された頭だし再生について説明するフローチャートである。ClipMark()のシンタクスは、図81に示すものとする。図1の記録再生装置1のブロック図を参照しながら説明する。

【0457】ステップS271において、制御部23は、Info.dvr, Clip Information file, PlayList fileおよびThumbnail fileの情報を取得し、ディスクに記録されているPlayListの一覧を示すGUI画面を作成し、ユーザーインターフェースを通して、GUIに表示する。

【0458】ステップS272において、ユーザーインターフェースを通して、ユーザにより1つのPlayListの再生が指示される。ステップS273において、制御部23は、PlayListMark、および、そのPlayListが参照するClipのClipMarkで参照されるピクチャから生成したサムネールのリストを、ユーザーインターフェースを通して、GUIに表示する。

【0459】ステップS274において、ユーザーインターフェースを通して、制御部23に、ユーザにより再生開始点のマーク点が指定される。ステップS275において、制御部23は、ステップS274における処理で選択されたマークがPlayListMarkにストアされているマークか否かを判断する。ステップS275において、選択されたマークがPlayListMarkにストアされているマークであると判断された場合、ステップS276に進み、ストアされていないマークであると判断された場合、ステップS278に進む。

【0460】ステップS276において、制御部23は、マークのPTSと、それが属するPlayItem_idを取得する。ステップS277において、制御部23はPlayItem_idが指すPlayItemが参照するAVストリームのSTC-sequence-idを取得する。

【0461】ステップS278において、制御部23は、STC-sequence-idとマークのPTSに基づいて、AVストリームをAVデコーダ27へ入力させる。具体的には、このSTC-sequence-idとマーク点のPTSを用いて、図132のフローチャートのステップS243, S244と同様の処理が行なわれる。ステップS279において、再生部3は、マーク点のPTSのピクチャから表示を開始する。

【0462】図9を参照して説明したように、PlayListが再生される時、そのPlayListが参照するClipのClipMa

rkにストアされているマークを参照する事ができる。従って、1つのClipを、Real PlayListや複数のVirtual PlayListによって参照している場合、それらのPlayListは、その1つのClipのClipMarkを共有する事ができるので、マークのデータを効率良く管理する事ができる。

【0463】仮に、ClipにClipMarkを定義しないで、PlayListだけにPlayListMarkとClipMarkを合わせたものを定義するようにした場合、上記の例のように1つのClipをReal PlayListや複数のVirtual PlayListによって参照している場合、それぞれのPlayListが同じ内容のClipのマーク情報を持つことになり、データの記録の効率が悪い。

【0464】図135は、PlayListMark()のシンタクスの別例を示す図である。lengthは、このlengthフィールドの直後のバイトからPlayListMark()の最後のバイトまでのバイト数を示す。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークのエントリー数を示す。

【0465】mark_invalid_flagは、1ビットのフラグであり、この値がゼロにセットされている時、このマークは有効な情報を持つことを示し、また、この値が1にセットされている時、このマークは無効であることを示す。

【0466】ユーザがユーザーインターフェース上で1つのマークのエントリーを消去するオペレーションをした時、記録再生装置1は、PlayListMarkからそのマークのエントリーを消去する代わりに、そのmark_invalid_flagの値を1に変更するようにしても良い。

【0467】mark_typeは、マークのタイプを示し、図136に示す意味を持つ。mark_name_lengthは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。このフィールドの値は32以下である。ref_to_PlayItem_idは、マークが置かれているところのPlayItemを指定するところのPlayItem_idの値を示す。あるPlayItemに対応するPlayItem_idの値は、PlayList()において定義される。

【0468】mark_time_stampは、そのマークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampは、ref_to_PlayItem_idで示されるPlayItemの中で定義されているところのIN_timeとOUT_timeで特定される再生範囲の中の時間を指す。タイムスタンプの意味は、図44と同じである。

【0469】entry_ES_PIDが、0xFFFFにセットされている場合、そのマークはPlayListによって使用されるすべてのエレメンタリーストリームに共通の時間軸上へのポインターである。entry_ES_PIDが、0xFFFFでない値にセットされている場合、entry_ES_PIDは、そのマークによって指されるところのエレメンタリーストリームを含んでいるところのトランスポートパケットのPIDの値を

示す。

【0470】ref_thumbnail_indexは、マークに付加されるサムネール画像の情報を示す。その意味は、図42のref_thumbnail_indexと同じである。mark_nameは、マークの名前を示す。このフィールドの中の左からmark_name_lengthで示されるバイト数が、有効なキャラクター文字であり、名前を示す。このキャラクター文字は、UIAppInfoPlayListの中でcharacter_setによって示される方法で符号化されている。

【0471】mark_nameフィールドの中で、それら有効なキャラクター文字に続くバイトの値は、どんな値が入っていても良い。このシンタクスの場合、マークが特定のエレメンタリーストリームを指すことができる。例えば、PlayListが、プログラムの中に複数のビデオストリームを持つマルチビュープログラムを参照している時、entry_ES_PIDは、そのプログラムの中の1つのビデオストリームを示すビデオPIDをセットする為に使われる。

【0472】ユーザがマルチビュープログラムを参照するところのPlayListを再生しており、そのユーザは、マルチビュー中の1つのビューを見ているとする。今、ユーザが記録再生装置1に対して、次のマーク点に再生をスキップするようにコマンドを送ったとする。この場合、記録再生装置1は、ユーザが現在見ているビューのビデオPIDと同じ値であるところのentry_ES_PIDのマークを使用するべきであり、記録再生装置1は、勝手にビューを変更すべきでない。記録再生装置1は、また、entry_ES_PIDが0xFFFFにセットされているマークを使用しても良い。この場合も記録再生装置1は、勝手にビューを変更しない。

【0473】図137は、図81に示すシンタクスのClipMark()の別例を示す図である。lengthは、このlengthフィールドの直後のバイトからClipMark()の最後のバイトまでのバイト数を示す。maker_IDは、mark_typeが0x60から0x7Fの値を示す時に、そのmark_typeを定義しているメーカーのメーカーIDを示す。

【0474】number_of_Clip_marksは、ClipMarkの中にストアされているマークのエントリー数を示す。mark_invalid_flagは、1ビットのフラグであり、この値がゼロにセットされている時、このマークは有効な情報を持つことを示し、また、この値が1にセットされている時、このマークは無効であることを示す。

【0475】ユーザが、ユーザーインターフェース上で1つのマークのエントリーを消去するオペレーションをした時、記録機はClipMarkからそのマークのエントリーを消去する代わりに、そのmark_invalid_flagの値が1に変更されるようにしても良い。mark_typeは、マークのタイプを示し、図138に示す意味を持つ。

【0476】ref_to_STC_idは、mark_time_stampとrepresentative_picture_time_stampの両方が置かれているところのSTC-sequenceを指定するところのSTC-sequence

-idを示す。STC-sequence-idの値は、STCInfo()の中で定義される。mark_time_stampは、図81のmark_entry()の場合でのmark_time_stampと同じ意味である。

【0477】entry_ES_PIDが、0xFFFFにセットされている場合、そのマークはClipの中のすべてのエレメンタリーストリームに共通の時間軸上へのポインターである。entry_ES_PIDが、0xFFFFでない値にセットされている場合、entry_ES_PIDは、そのマークによって指されるところのエレメンタリーストリームを含んでいるところのトランスポートパケットのPIDの値を示す。

【0478】ref_to_thumbnail_indexは、マークに付加されるサムネール画像の情報を示す。その意味は、図78のref_thumbnail_indexと同じである。representative_picture_time_stampは、図81のrepresentative_picture_entry()の場合でのmark_time_stampと同じ意味である。

【0479】図137に示したシンタクスの場合、マークが、特定のエレメンタリーストリームを指すことができる。例えば、Clipが、プログラムの中に複数のビデオストリームを持つマルチビュープログラムを含んでいるとき、entry_ES_PIDは、そのプログラムの中の1つのビデオストリームを示すビデオPIDをセットする為に使われる。

【0480】ユーザが、マルチビュープログラムを参照するところのPlayListを再生しており、そのユーザは、マルチビュー中の1つのビューを見ているとする。今、ユーザが記録再生装置1に対して、次のマーク点に再生をスキップするようにコマンドを送ったとする。この場合、記録再生装置1は、ユーザが現在見ているビューのビデオPIDと同じ値であるところのentry_ES_PIDのマークを使用するべきであり、記録再生装置1は、勝手にビューを変更すべきでない。記録再生装置1は、また、entry_ES_PIDが0xFFFFにセットされているマークを使用しても良い。この場合も記録再生装置1は、勝手にビューを変更しない。

【0481】このようなシンタクス、データ構造、規則に基づく事により、記録媒体100に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが、再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

【0482】本実施の形態のデータベース構成によれば、PlayListファイルやClip Informationファイルを別々に分離して記録するので、編集などによって、所定のPlayListやClipの内容が変更されたとき、そのファイルに関係のない他のファイルを変更する必要がない。従つて、ファイルの内容の変更が容易に行え、またその変更および記録にかかる時間を小さくできる。

【0483】また、最初にInfo.dvrだけを読み出して、ディスクの記録内容をユーザーインターフェースへ提示

し、ユーザが再生指示したPlayListファイルと、それに関連するClip Informationファイルだけをディスクから読み出すようにすれば、ユーザの待ち時間を小さくすることができる。

【0484】仮に、すべてのPlayListファイルやClip Informationファイルを1つのファイルにまとめて記録すると、そのファイルサイズは非常に大きくなる。そのために、そのファイルの内容を変更して、それを記録するためにかかる時間は、個々のファイルを別々に分離して記録する場合に比べて、非常に大きくなる。本発明を適用することにより、このようなことを防ぐことが可能となる。

【0485】上述したように、AVストリームファイル、すなわちClip AVストリームファイルの中の特徴的な画像を指示するマークをストアするClipMarkを、前記AVストリームの管理情報データファイル、すなわちClip Informationファイルに記録し、また、AVストリーム中の指定された区間の組み合わせにより定義される1つの再生手順の情報を持つオブジェクト、すなわちPlayListの再生範囲の中から、ユーザが指定したマーク点、または、そのオブジェクトを再生するときのResume点を示すマークをストアするPlayListMarkを、オブジェクトに記録する。

【0486】このようにすることにより、PlayListが再生される時、そのPlayListが参照するClipのClipMarkにストアされているマークを参照する事ができる。従つて、1つのClipをReal PlayListや複数のVirtual PlayListによって参照している場合、それらのPlayListは、その1つのClipのClipMarkを共有することができるので、マークのデータを効率良く管理することができる。

【0487】仮に、ClipにClipMarkを定義しないで、PlayListだけにPlayListMarkとClipMarkを合わせたものを定義するようにした場合、上記の例のように1つのClipをReal PlayListや複数のVirtual PlayListによって参照している場合、それぞれのPlayListが同じ内容のClipのマーク情報を持つことになり、データの記録の効率が悪い。本発明を適用することにより、このようなことを防ぐことが可能となる。

【0488】以上のように、AVストリームの付属情報として、エントリーポイントのアドレスをストアするためのEP_mapと、マーク点のピクチャのタイプ（例えば番組の頭出し点）とそのピクチャのAVストリームの中のアドレスをストアするためのClipMarkを、Clip Information Fileとしてファイル化して記録媒体100に記録することにより、AVストリームの再生に必要なストリームの再生に必要なストリームの符号化情報を適切に管理することが可能である。

【0489】このClip Information file情報により、ユーザが、記録媒体100に記録されているAVストリームの中から興味のあるシーン、例えば番組の頭出し点な

ど、をサーチすることができ、ユーザのランダムアクセスや特殊再生の指示に対して、記録媒体100からのAVストリームの読み出し位置の決定が容易になり、またストリームの復号開始を速やかに行うことができる。

【0490】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。この場合、例えば、記録再生装置1は、図139に示されるようなパーソナルコンピュータにより構成される。

【0491】図139において、CPU(Central Processing Unit)201は、ROM(Read Only Memory)202に記憶されているプログラム、または記憶部208からRAM(Random Access Memory)203にロードされたプログラムに従って各種の処理を実行する。RAM203にはまた、CPU201が各種の処理を実行する上において必要なデータなども適宜記憶される。

【0492】CPU201、ROM202、およびRAM203は、バス204を介して相互に接続されている。このバス204にはまた、入出力インターフェース205も接続されている。

【0493】入出力インターフェース205には、キーボード、マウスなどよりなる入力部206、CRT、LCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部207、ハードディスクなどより構成される記憶部208、モ뎀、ターミナルアダプタなどより構成される通信部209が接続されている。通信部209は、ネットワークを介しての通信処理を行う。

【0494】入出力インターフェース205にはまた、必要に応じてドライブ210が接続され、磁気ディスク221、光ディスク222、光磁気ディスク223、或いは半導体メモリ224などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部208にインストールされる。

【0495】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

【0496】この記録媒体は、図139に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク221(フロッピディスクを含む)、光ディスク222(CD-ROM(Compact Disk-Read Only Memory)、DVD(Digital Versatile Disk)を含む)、光磁気ディスク223(MD(Mini-Disk)を含む)、若しくは半導体メモリ224などよりなるパッケージメディアにより構

成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM202や記憶部208が含まれるハードディスクなどで構成される。

【0497】なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0498】また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【0499】

【発明の効果】以上のように本発明の第1の情報処理装置および方法、並びにプログラムにおいては、入力されたAVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを、AVストリームを管理するための管理情報として生成するとともに、AVストリーム中の所定の区間の組み合わせを定義するPlaylistに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlaylistMarkを生成し、ClipMark、およびPlaylistMarkを各々独立したテーブルとして記録媒体に記録するようにしたので、AVストリームの所望の位置に、迅速且つ確実にアクセスすることが可能となる。

【0500】また本発明の第2の情報処理装置および方法、並びにプログラムは、AVストリームから抽出された特徴的な画像を指し示すマークで構成されるClipMarkを含むAVストリームを管理するための管理情報と、AVストリーム中の所定の区間の組み合わせを定義するPlaylistに対応する再生区間の中から、ユーザが任意に指定した画像を指し示すマークから構成されるPlaylistMarkを読み出し、その読み出された管理情報とPlaylistMarkによる情報を提示し、提示された情報から、ユーザが再生を指示したPlaylistに対応するClipMarkを参照し、参照されたClipMarkを含み、ClipMarkに対応する位置からAVストリームを再生するようにしたので、AVストリームの所望の位置に、迅速且つ確実にアクセスすることが可能となる。

【図面の簡単な説明】

【図1】本発明を適用した記録再生装置の一実施の形態の構成を示す図である。

【図2】記録再生装置1により記録媒体に記録されるデータのフォーマットについて説明する図である。

【図3】Real PlaylistとVirtual Playlistについて説明する図である。

【図4】Real Playlistの作成について説明する図である。

【図5】Real Playlistの削除について説明する図である。

【図6】アセンブル編集について説明する図である。

【図7】Virtual PlayListにサブパスを設ける場合について説明する図である。

【図8】PlayListの再生順序の変更について説明する図である。

【図9】PlayList上のマークとClip上のマークについて説明する図である。

【図10】メニュー・サムネイルについて説明する図である。

【図11】PlayListに付加されるマークについて説明する図である。

【図12】クリップに付加されるマークについて説明する図である。

【図13】PlayList、Clip、サムネイルファイルの関係について説明する図である。

【図14】ディレクトリ構造について説明する図である。

【図15】info.dvrのシンタクスを示す図である。

【図16】DVR volumeのシンタクスを示す図である。

【図17】Resumevolumeのシンタクスを示す図である。

【図18】UIAppInfovolumeのシンタクスを示す図である。

【図19】Character set valueのテーブルを示す図である。

【図20】TableOfPlayListのシンタクスを示す図である。

【図21】TableOfPlayListの他のシンタクスを示す図である。

【図22】MakersPrivateDataのシンタクスを示す図である。

【図23】xxxxx.rplsとyyyyy.vplsのシンタクスを示す図である。

【図24】PlayListについて説明する図である。

【図25】PlayListのシンタクスを示す図である。

【図26】PlayList_typeのテーブルを示す図である。

【図27】UIAppInfoPlayListのシンタクスを示す図である。

【図28】図27に示したUIAppInfoPlayListのシンタクス内のフラグについて説明する図である。

【図29】PlayItemについて説明する図である。

【図30】PlayItemについて説明する図である。

【図31】PlayItemについて説明する図である。

【図32】PlayItemのシンタクスを示す図である。

【図33】IN_timeについて説明する図である。

【図34】OUT_timeについて説明する図である。

【図35】Connection_Conditionのテーブルを示す図である。

【図36】Connection_Conditionについて説明する図である。

【図37】BridgeSequenceInfoを説明する図である。

【図38】BridgeSequenceInfoのシンタクスを示す図である。

【図39】SubPlayItemについて説明する図である。

【図40】SubPlayItemのシンタクスを示す図である。

【図41】SubPath_typeのテーブルを示す図である。

【図42】PlayListMarkのシンタクスを示す図である。

【図43】Mark_typeのテーブルを示す図である。

【図44】Mark_time_stampを説明する図である。

【図45】zzzzz.clipのシンタクスを示す図である。

【図46】ClipInfoのシンタクスを示す図である。

【図47】Clip_stream_typeのテーブルを示す図である。

【図48】offset_SPNについて説明する図である。

【図49】offset_SPNについて説明する図である。

【図50】STC区間について説明する図である。

【図51】STC_Infoについて説明する図である。

【図52】STC_Infoのシンタクスを示す図である。

【図53】ProgramInfoを説明する図である。

【図54】ProgramInfoのシンタクスを示す図である。

【図55】VideoCodingInfoのシンタクスを示す図である。

【図56】Video_formatのテーブルを示す図である。

【図57】frame_rateのテーブルを示す図である。

【図58】display_aspect_ratioのテーブルを示す図である。

【図59】AudioCodingInfoのシンタクスを示す図である。

【図60】audio_codingのテーブルを示す図である。

【図61】audio_component_typeのテーブルを示す図である。

【図62】sampling_frequencyのテーブルを示す図である。

【図63】CPIについて説明する図である。

【図64】CPIについて説明する図である。

【図65】CPIのシンタクスを示す図である。

【図66】CPI_typeのテーブルを示す図である。

【図67】ビデオEP_mapについて説明する図である。

【図68】EP_mapについて説明する図である。

【図69】EP_mapについて説明する図である。

【図70】EP_mapのシンタクスを示す図である。

【図71】EP_type valuesのテーブルを示す図である。

【図72】EP_map_for_one_stream_PIDのシンタクスを示す図である。

【図73】TU_mapについて説明する図である。

【図74】TU_mapのシンタクスを示す図である。

【図75】ClipMarkのシンタクスを示す図である。

【図76】mark_typeのテーブルを示す図である。

【図77】mark_type_stampのテーブルを示す図である。

【図78】ClipMarkのシンタクスの他の例を示す図である。

る。

【図79】Mark_typeのテーブルの他の例を示す図である。

【図80】mark_entry()とrepresentative_picture_entry()の例を示す図である。

【図81】mark_entry()とrepresentative_picture_entry()のシンタクスを示す図である。

【図82】mark_entry()とrepresentative_picture_entry()のシンタクスの他の例を示す図である。

【図83】RSPN_ref_EP_startとoffset_num_picturesの関係を説明する図である。

【図84】mark_entry()とrepresentative_picture_entry()のシンタクスの他の例を示す図である。

【図85】ClipMarkとEP_mapの関係を説明する図である。

【図86】menu.thmbとmark.thmbのシンタクスを示す図である。

【図87】Thumbnailのシンタクスを示す図である。

【図88】thumbnail_picture_formatのテーブルを示す図である。

【図89】tn_blockについて説明する図である。

【図90】DVR MPEG2のトランスポートストリームの構造について説明する図である。

【図91】DVR MPEG2のトランスポートストリームのレコーダモデルを示す図である。

【図92】DVR MPEG2のトランスポートストリームのプレーヤモデルを示す図である。

【図93】source packetのシンタクスを示す図である。

【図94】TP_extra_headerのシンタクスを示す図である。

【図95】copy permission indicatorのテーブルを示す図である。

【図96】シームレス接続について説明する図である。

【図97】シームレス接続について説明する図である。

【図98】シームレス接続について説明する図である。

【図99】シームレス接続について説明する図である。

【図100】シームレス接続について説明する図である。

【図101】オーディオのオーバーラップについて説明する図である。

【図102】BridgeSequenceを用いたシームレス接続について説明する図である。

【図103】BridgeSequenceを用いないシームレス接続について説明する図である。

【図104】DVR STDモデルを示す図である。

【図105】復号、表示のタイミングチャートを示す図である。

【図106】図81のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図107】図81のシンタクスの場合における再生の動作を説明する図である。

【図108】EP_mapの例を示す図である。

【図109】ClipMarkの例を示す図である。

【図110】図81のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

【図111】図81のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

【図112】図82のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図113】図82のシンタクスの場合における再生を説明する図である。

【図114】EP_mapの例を示す図である。

【図115】ClipMarkの例を示す図である。

【図116】図82のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図117】図82のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図118】図84のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図119】図84のシンタクスの場合における再生を説明する図である。

【図120】EP_mapの例を示す図である。

【図121】ClipMarkの例を示す図である。

【図122】図84のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図123】図84のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図124】アプリケーションフォーマットを示す図である。

【図125】PlayList上のマークとClip上のマークを説明する図である。

【図126】ClipMarkのシンタクスの他の例を示す図である。

【図127】ClipMarkのシンタクスのさらに他の例を示す図である。

【図128】アナログAV信号をエンコードして記録する場合のClipMarkの作成について説明するフローチャートである。

【図129】トランスポートストリームを記録する場合のClipMarkの作成について説明するフローチャートである。

【図130】RealPlaylistの作成について説明するフローチャートである。

【図131】VirtualPlaylistの作成について説明するフローチャートである。

【図132】Playlistの再生について説明するフローチャートである。

【図133】PlayListMarkの作成について説明するフローチャートである。

【図134】PlayListを再生する際の頭出し再生について説明するフローチャートである。

【図135】PlayListMarkのシンタクスを示す図である。

【図136】PlayListMarkのMark_typeを説明するための図である。

【図137】ClipMarkの他のシンタクスを示す図である。

【図138】ClipMarkのMark_typeを説明するための図

である。

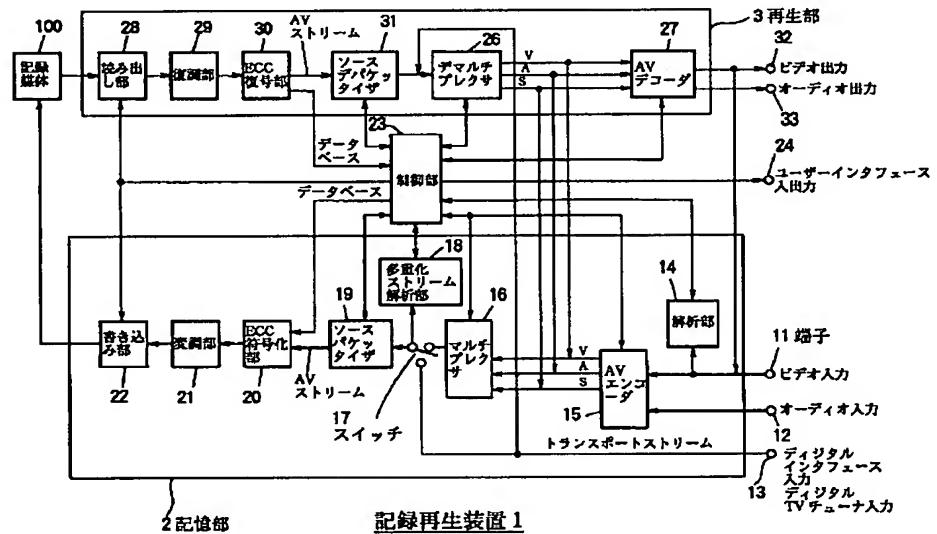
[図1]

【図139】媒体を説明する図である。

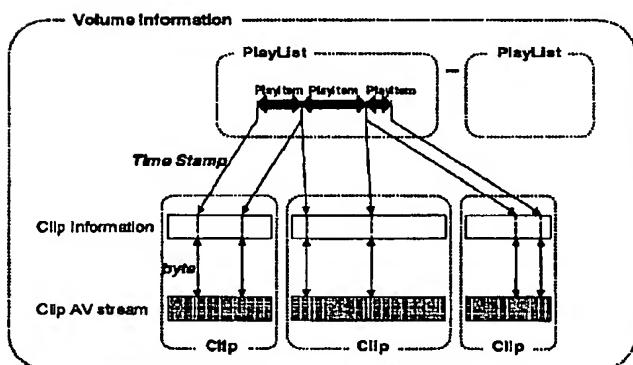
【符号の説明】

1 記録再生装置, 11 乃至 13 端子, 14 解析部, 15 AVエンコーダ, 16 マルチブレクサ, 17 スイッチ, 18 多重化ストリーム解析部, 19 ソースパケッタタイザ, 20 ECC符号化部, 21 変調部, 22 書き込み部, 23 制御部, 24 ユーザインタフェース, 25 スイッチ, 26 デマルチブレクサ, 27 AVデコーダ, 28 読み出し部, 29 復調部, 30 ECC復号部, 31 ソースパケッタタイザ, 32, 33 端子

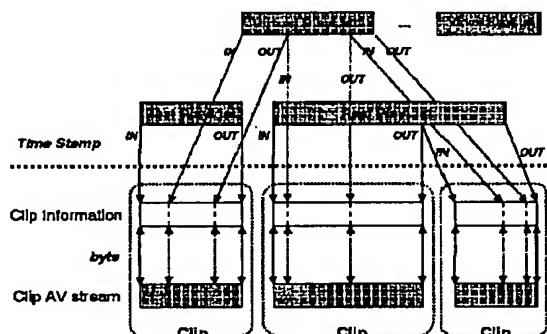
〔図1〕



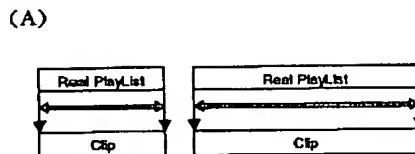
[図2]



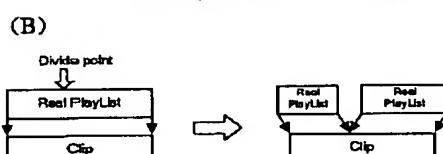
[图3]



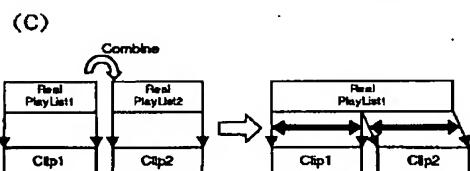
【図4】



Real PlayList のクリエイトの例



Real PlayList のディバイドの例

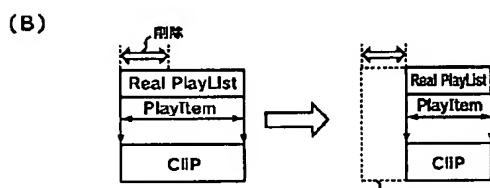


Real PlayList のコンパインの例

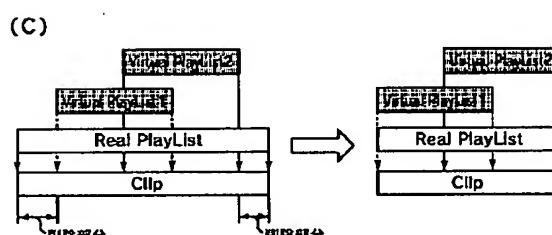
【図5】



Real PlayList 全体のテリートの例

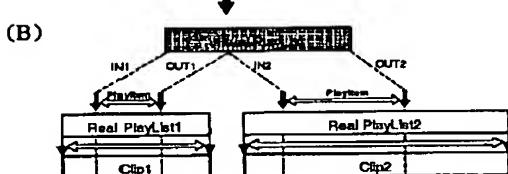
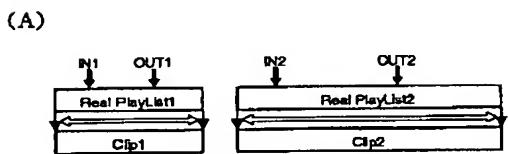


Real PlayList の部分的なテリートの例



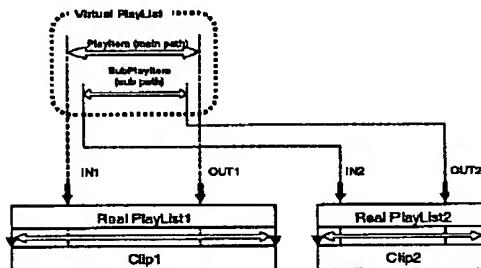
Real PlayList のミニマイズの例

【図6】



アセンブル組集の例

【図7】



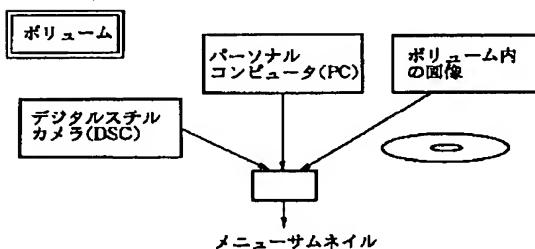
Virtual PlayList へのオーディオのアフレコの例

【図19】

Value	Character coding
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xf	Reserved

Character set value

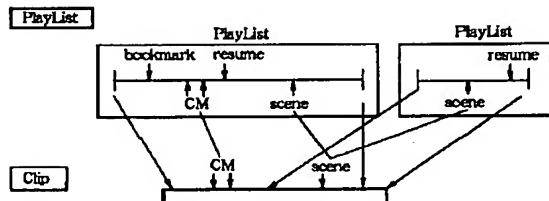
【図10】



〔图8〕



[图9]



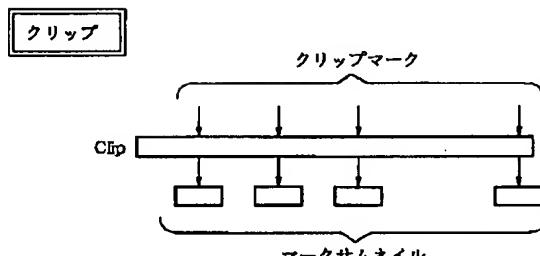
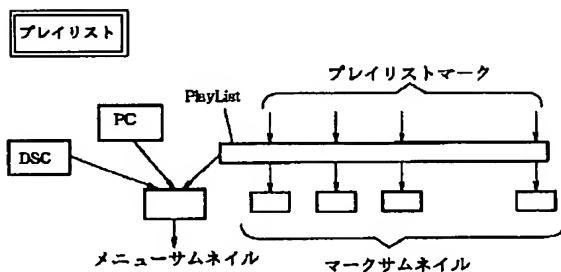
【图26】

PlayList_type	Meaning
0	AV記録のための PlayList この PlayList に参照されるすべての Clip は、一つ以上のビデオストリームを含まなければならない。
1	オーディオ記録のための PlayList この PlayList に参照されるすべての Clip は、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2-255	reserved

PlayList_type

〔図11〕

【图 12】

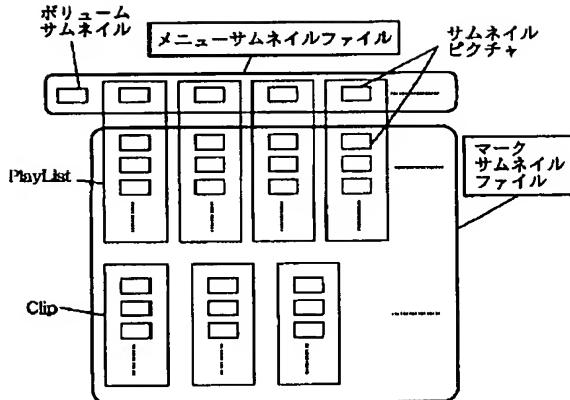


【図16】

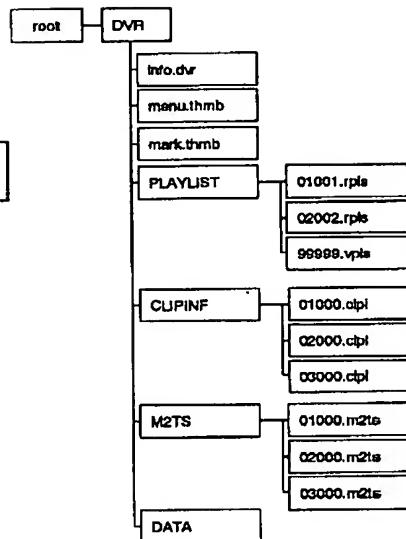
Syntax	No. of bits	Mnemonics
DVRVolume()		
version_number	8*4	bstbf
length	32	tlmsbf
ResumeVolume()		
UnappingVolume()		
}		

DVR Volume のシンタクス

【図13】



【図14】



【図108】

EP_map	
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

【図114】

EP_map	
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

【図15】

Syntax	No. of	Mnemonics
info.drv {		
TableOfPlayLists_Start_address	32	ulmsbf
MakersPrivateData_Start_address	32	ulmsbf
reserved	162	bslbf
DVRVolume()		
for(=0;j<N1;j++) {		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for(=0;j<N2;j++) {		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

info.drv のシンタクス

【図18】

Syntax	No. of	Mnemonics
UIAppInfoVolume () {		
character_set	8	bslbf
name_length	8	ulmsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	ulmsbf
reserved_for_future_use	256	bslbf
}		

UIAppInfoVolume のシンタクス

Syntax	No. of	Mnemonics
xxxxxx.rpls / yyyy.yvpls {		
PlayListMark_Start_address	32	ulmsbf
MakersPrivateData_Start_address	32	ulmsbf
reserved	162	bslbf
PlayList()		
for(=0;j<N1;j++) {		
padding_word	16	bslbf
}		
PlayListMark()		
for(=0;j<N2;j++) {		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

ResumeVolume のシンタクス

【図23】

xxxxxx.rpls と yyyy.yvpls のシンタクス

【図66】

CPI_type	Meaning
0	EP_map type
1	TU_map type

CPI_type の意味

【図20】

Syntax	No. of bits	Mnemonics
TableOfPlayLists()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++) {		
PlayList file_name	8*10	bslbf
}		
}		

【図28】

(A)

write_protect_flag	Meaning
0b	その PlayList を自由に消去しても良い。
1b	write_protect_flag を除いてその PlayList の内容は、消去および変更されるべきではない。

write_protect_flag

(B)

TableOfPlayLists のシンタクス

【図21】

■ TableOfPlayLists - シンタクス (4.2.3.2 の別案)

Syntax	No. of bits	Mnemonics
TableOfPlayLists()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++) {		
PlayList file_name	8*10	bslbf
UIAppInfoPlayList0		
}		
}		

is_played_flag	Meaning
0b	その PlayList は、記録されてから一度も再生されたことがない。
1b	PlayList は、記録されてから一度は再生された。

is_played_flag

(C)

archive	Meaning
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

archive

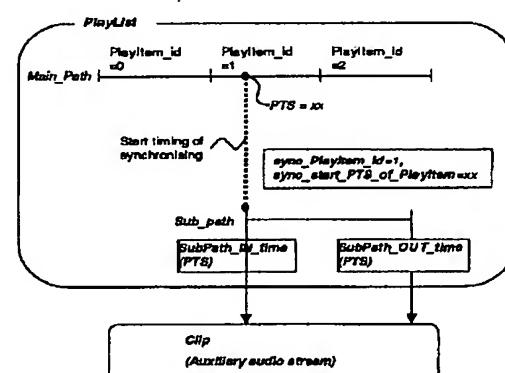
TableOfPlayLists の別シンタクス

【図39】

【図22】

Syntax	No. of bits	Mnemonics
MakersPrivateData()		
version_number	8*4	bslbf
length	32	uimsbf
if (length != 0) {		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_maker_entries; i++) {		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for (i=0; i<number_of_mpd_blocks; i++) {		
mpd_block	mpd_block_size=1024b	
}		
}		
}		

MakersPrivateData のシンタクス

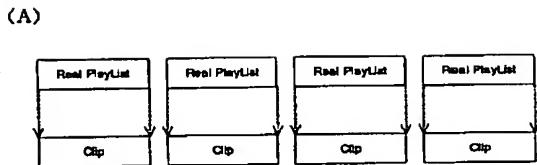


【図41】

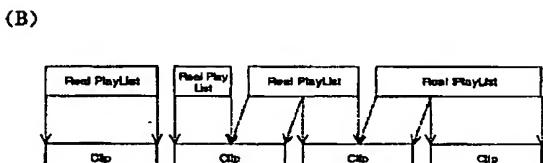
SubPath_type	Meaning
0x00	Auxiliary audio stream path
0x01 - 0xff	reserved

SubPath_type

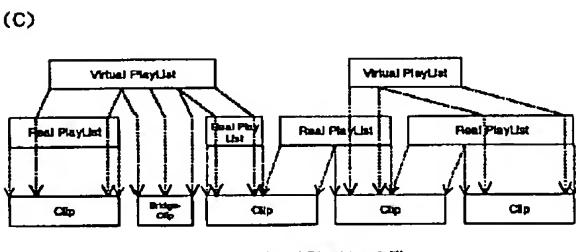
【図24】



初めて AV ストリームが Clip として記録された時の Real Playlist の例



図集後の Real Playlist の例



Virtual Playlist の例

【図25】

Syntax	No. of bits	Mnemonics
PlayList()		
version_number	8*4	bslbf
length	32	ulmsbf
PlayList type	8	ulmsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	ulmsbf
if (<Virtual PlayList>) {		
number_of_SubPlayItems // sub path	16	ulmsbf
} else {		
reserved	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++) {		
PlayItem()		
}		
if (<Virtual PlayList>) {		
if (CPI_type==0 && PlayList_type==0) {		
for (i=0; i<number_of_SubPlayItems; i++)		
SubPlayItem() // sub path		
}		
}		
}		

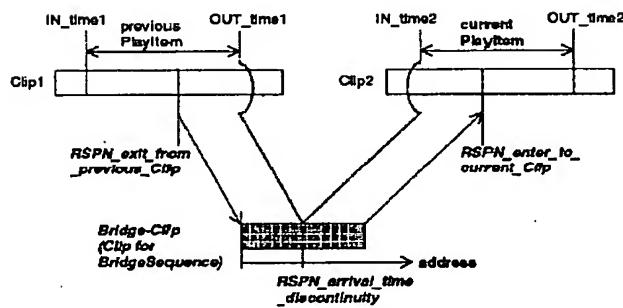
PlayList のシンタクス

【図33】

CPI_type	In the PlayList()	Semantics of IN_time
EP_map type		IN_time は、PlayItem の中で最初のプレゼンテーションユニットに対する33ビットのPTSの上位32ビットを示さなければならない。
TU_map type		IN_time は、TU_map_time_axis 上の時刻でなければならない。かつ、IN_time は、time_unit の精度に丸めて表さなければならない。IN_time は、次に示す等式により計算される。 $IN_time = TU_start_time \% 2^{32}$

IN_time

【図37】

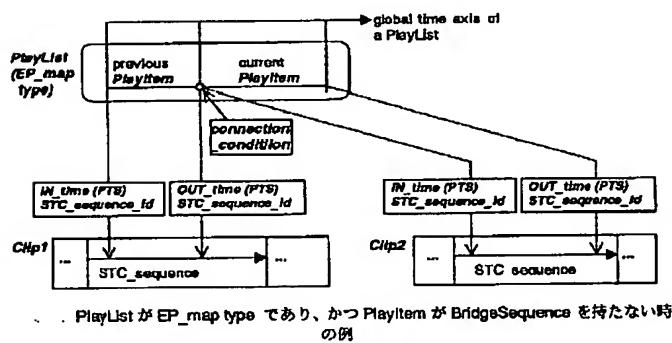


【図27】

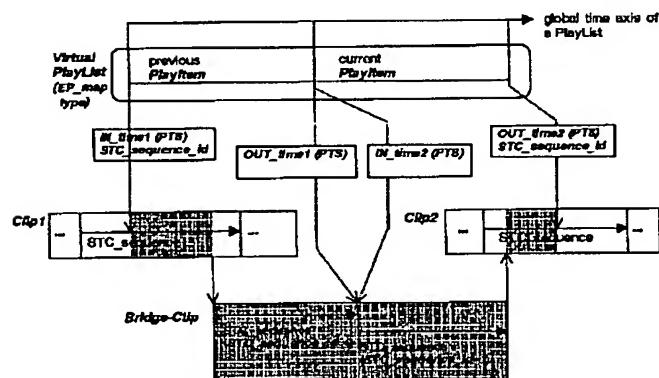
Syntax	No. of bits	Mnemonics
UIAppInfoPlayList2()		
character_set	8	bslbf
name_length	8	ulmsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*8	bslbf
valid_period	4*8	bslbf
maker_id	16	ulmsbf
maker_code	16	ulmsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	ulmsbf
reserved for future use	256	bslbf

UIAppInfoPlayList のシンタクス

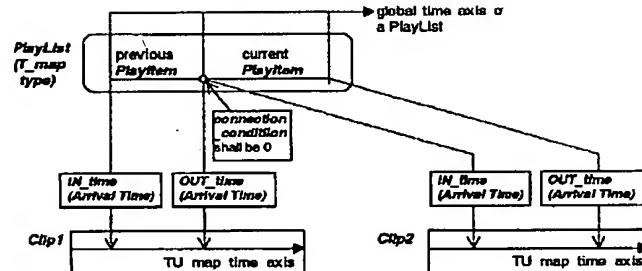
【図29】



【図30】



【図31】



Playlist が TU_map type である時の例

【図32】

Syntax	No. of bits	Mnemonics
PlayItem0 {		
Clip information file name	8*10	bslbf
reserved	24	bslbf
STC sequence Id	8	ulmsbf
IN_time	32	ulmsbf
OUT_time	32	ulmsbf
reserved	14	bslbf
connection condition	2	bslbf
if (<Virtual Playlist>){		
if (connection_condition=='10') {		
BridgeSequenceInfo()		
}		
}		
}		

PlayItem のシンタクス

【図34】

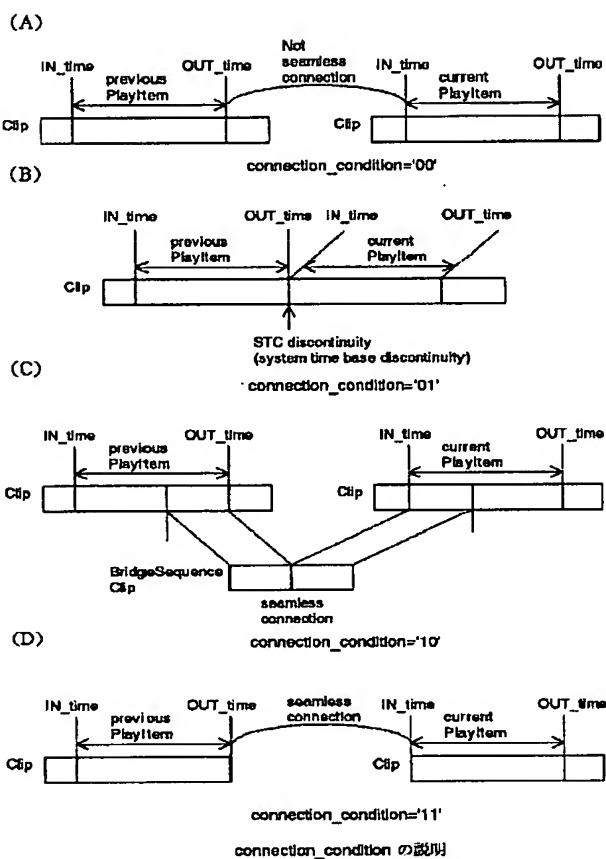
CPI_type in the Playlist0	Semantics of OUT_time
EP_map type	OUT_time は、次に示す等式によって計算される Presentation_end_TS の値の上位 32 ビットを示さなければならない。 $\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$ ここで、 PTS_out は、PlayItem の中で最後のプレゼンテーションユニットに対応する 33 ビット長の PTS である。 AU_duration は、最後のプレゼンテーションユニットの 80kHz 単位の表示期間である。
TU_map type	OUT_time は、TU_map_time_axis 上の時刻でなければならない。かつ、OUT_time は、time_unit の精度に丸めて表さねばならない。 OUT_time は、次に示す等式により計算される。 $\text{OUT_time} = \text{TU_start_time \% } 2^{32}$

OUT_time

【図35】

connection condition	meaning
00	<ul style="list-style-type: none"> 先行する PlayItem と現在の PlayItem の接続は、シームレス再生の保証がなされていない。 PlayList の CPI_type が TU_map type である場合、connection_condition は、この値をセットされねばならない。
01	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 先行する PlayItem と現在の PlayItem は、システムタイムベース (STC ベース) の不連続点があるために分割されていることを表す。
10	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 この状態は、Virtual PlayList に対してだけ許される。 先行する PlayItem と現在の PlayItem との接続は、シームレス再生の保証がなされている。 先行する PlayItem と現在の PlayItem は、BridgeSequence を使用して接続されており、DVR MPEG-2 トランスポортストリームは、後述する DVR-STD に従っていなければならない。
11	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 先行する PlayItem と現在の PlayItem は、シームレス再生の保証がなされている。 先行する PlayItem と現在の PlayItem は、BridgeSequence を使用しないで接続されており、DVR MPEG-2 トランスポортストリームは、後述する DVR-STD に従っていなければならぬ。

【図36】



【図38】

Syntax	No. of bits	Mnemonics
BridgeSequenceInfo() {		
Bridge Clip Information file name	8~10	bsbf
RSPN exit from previous Clip	32	uimstb
RSPN enter to current Clip	32	uimstb
}		

BridgeSequenceInfo のシンタクス

【図47】

Clip_stream_type	meaning
0	Clip AV ストリーム
1	Bridge-Clip AV ストリーム
2 - 255	Reserved

Clip_stream_type

【図40】

Syntax	No. of bits	Mnemonics
SubPlayItem()		
Clip Information file name	8*10	belbf
SubPath type	8	belbf
sync PlayItem Id	8	uimsbf
sync start PTS of PlayItem	32	uimsbf
SubPath IN time	32	uimsbf
SubPath OUT time	32	uimsbf
}		

【図56】

video_format	Meaning
0	480i
1	576i
2	480p (including 640x480p format)
3	1080i
4	720p
5	1080p
6 - 254	reserved
255	No information

video_format

SubPlayItem のシンタクス

【図42】

Syntax	No. of bits	Mnemonics
PlayListMark()		
version number	8*4	belbf
length	32	uimsbf
number of PlayList marks	16	uimsbf
for(i=0; i < number of PlayList marks; i++) {		
reserved	8	belbf
mark type	8	belbf
mark time stamp	32	uimsbf
PlayItem Id	8	uimsbf
reserved	24	uimsbf
character set	8	belbf
name length	8	uimsbf
mark name	8*256	belbf
ref.thumbnail index	16	uimsbf
}		
}		

PlayListMark のシンタクス

【図43】

Mark type	Meaning	Comments
0x00	resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	book-mark	PlayList の再生エントリーポイント。このマークは、ユーザがセトすることができ、例えば、お気に入りのシーンの開始点を指定するマークを使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログラムの最後まで、プレーヤはプログラムをスキップする。PlayListMark()において定義されるスキップマークポイントの数は、0または1でなければならない。
0x03 - 0xBF	reserved	
0x90 - 0xFF	reserved	Reserved for ClipMark()

mark_type

【図44】

CPI_type_in the Playlist()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならぬ。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならぬ。かつ、mark_time_stamp は、time_unit の精度に丸めて表さねばならない。mark_time_stamp は、次に示す式により計算される。 $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

mark_time_stamp

【図46】

Syntax	No. of bits	of	Mnemonics
zzzzz.cpl{			
STC_Info_Start_address	32	ulmsbf	
ProgramInfo_Start_address	32	ulmsbf	
CPI_Start_address	32	ulmsbf	
ClipMark_Start_address	32	ulmsbf	
MakersPrivateData_Start_address	32	ulmsbf	
reserved	86	bsbf	
ClipInfo()			
for(i=0;i<n1;i++){			
padding_word	16	bsbf	
}			
STC_Info()			
for(i=0;i<n2;i++){			
padding_word	16	bsbf	
}			
ProgramInfo()			
for(i=0;i<n3;i++){			
padding_word	16	bsbf	
}			
CPI()			
for(i=0;i<n4;i++){			
padding_word	16	bsbf	
}			
ClipMark()			
for(i=0;i<n5;i++){			
padding_word	16	bsbf	
}			
MakersPrivateData()			
}			

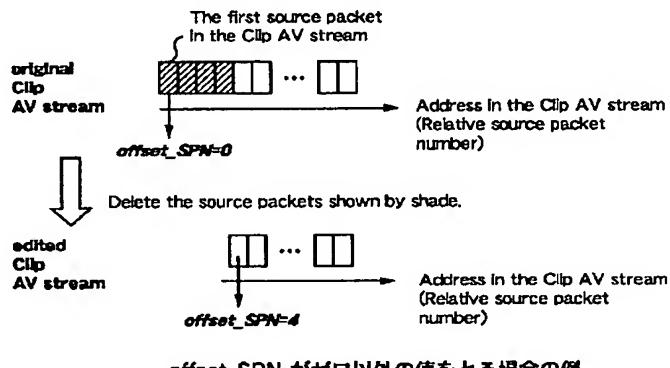
【図45】

Syntax	No. of bits	of	Mnemonics
ClipInfo{			
version_number	8*4	bsbf	
length	32	ulmsbf	
Clip_stream_type	8	bsbf	
offset_SPN	32	ulmsbf	
TS_recording_rate	24	ulmsbf	
reserved	8	bsbf	
record_time_and_date	4*14	bsbf	
reserved	8	bsbf	
duration	4*8	bsbf	
reserved	7	bsbf	
time_controlled_flag	1	bsbf	
TS_average_rate	24	ulmsbf	
//Clip_stream_type==1 //Bridge-Clip AV stream			
RSPN_arrival_time_discontinuity	32	ulmsbf	
else			
reserved	32	bsbf	
reserved_for_system_use	144	bsbf	
reserved	11	bsbf	
is_formed_identifier_valid	1	bsbf	
is_original_network_ID_valid	1	bsbf	
is_transport_stream_ID_valid	1	bsbf	
is_service_ID_valid	1	bsbf	
is_country_code_valid	1	bsbf	
format_identifier	32	bsbf	
original_network_ID	16	ulmsbf	
transport_stream_ID	16	ulmsbf	
service_ID	16	ulmsbf	
country_code	24	bsbf	
stream_format_name	16*8	bsbf	
reserved_for_future_use	256	bsbf	
}			

ClipInfo のシンタクス

zzzzz.cpl のシンタクス

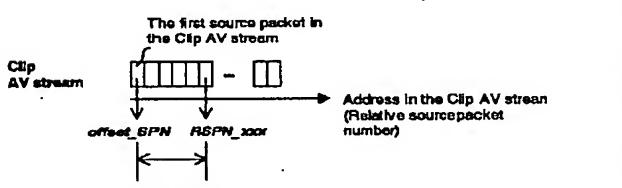
【図48】



offset_SPN がゼロ以外の値をとる場合の例

【図57】

【図49】

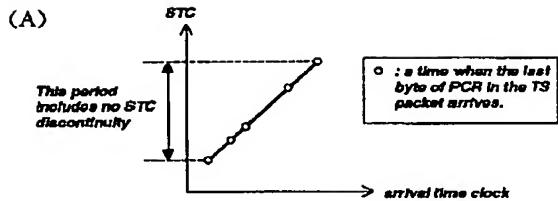


AVストリームでの offset_SPN と相対ソースパケット番号 (RSPN_xxx) の間の関係

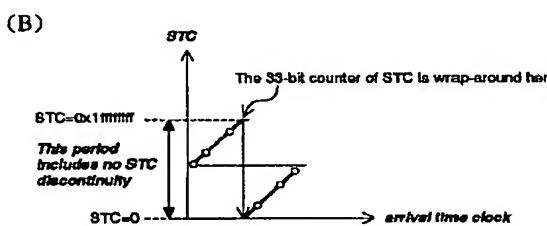
frame_rate	Meaning
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97...)
5	30
6	50
7	60 000/1001 (59.94 ...)
8	60
9 - 254	reserved
255	No Information

frame_rate

【図50】

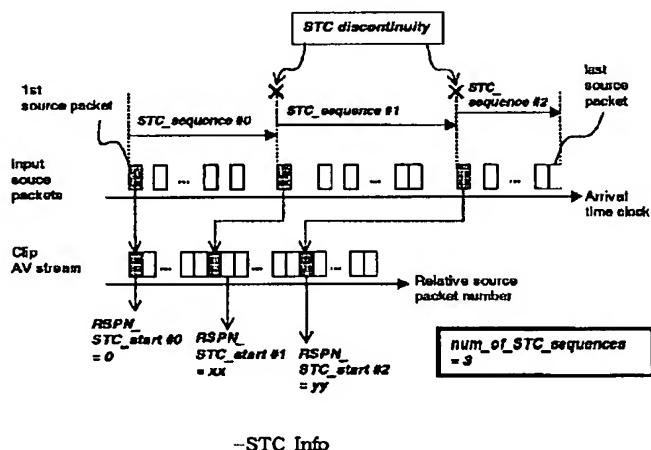


Syntax	No. of bits	Mnemonics
AudioCodingInfo {		
audio_coding	8	ulmsbf
audio_component_type	8	ulmsbf
sampling_frequency	8	ulmsbf
reserved	8	bslbf
}		



AudioCodingInfo のシンタクス

【図51】



【図52】

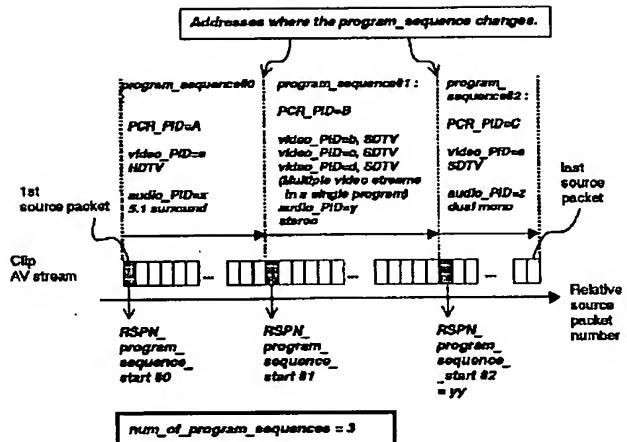
Syntax	No. of bits	Mnemonics
STC_Info {		
version_number	8*4	bslbf
length	32	ulmsbf
if (length != 0) {		
reserved	8	bslbf
num_of_STC_sequences	8	ulmsbf
for(STC_sequence_id=0;		
STC_sequence_id < num_of_STC_sequences;		
STC_sequence_id++) {		
reserved	32	bslbf
RSPN_STC_start	32	ulmsbf
}		
}		
}		
}		

STC_Info のシンタクス

audio_component_type	Meaning
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-264	reserved
255	No Information

audio_component_type

【図53】



【図58】

ProgramInfo の例

【図54】

Syntax	No. of bits	Mnemonics
ProgramInfo()		
version_number	8*4	bslbf
length	32	uimbf
if (length != 0) {		
reserved	8	bslbf
number_of_program_sequences	8	uimbf
for (i=0; i<number_of_program_sequences; i++) {		
RSPN_program_sequence_start	32	uimbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimbf
number_of_audios	8	uimbf
for (k=0; k<number_of_videos; k++) {		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0; k<number_of_audios; k++) {		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		

ProgramInfo のシンタクス

【図55】

Syntax	No. of bits	Mnemonics
VideoCodingInfo()		
video_format	8	uimbf
frame_rate	8	uimbf
display_aspect_ratio	8	uimbf
reserved	8	bslbf

VideoCodingInfo のシンタクス

【図62】

sampling_frequency	Meaning
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

sampling_frequency

【図60】

audio_coding	Meaning
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No Information

audio_coding

【図65】

Syntax	No. of bits	Mnemonics
CPI0 {		
version_number	8*4	bsb1
length	32	uimsbf
reserved	15	bsb1
CPI_type	1	bsb1
if (CPI_type == 0)		
EP_map0		
else		
TU_map0		
}		

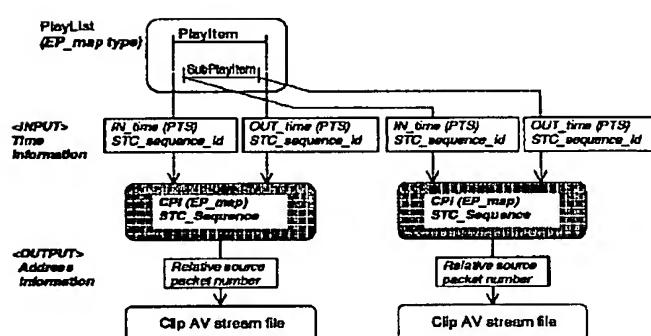
CPI のシンタクス

【図76】

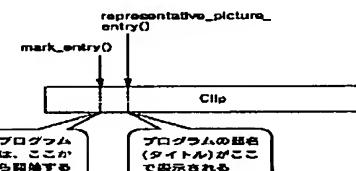
Mark_type	Meaning	Comments
0x00 - 0x8F	reserved	Reserved for PlaylistMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点。
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点。
0x92	Scene-start mark	シーンチェンジポイントを示すマーク。
0x93 - 0xFF	reserved	

mark_type

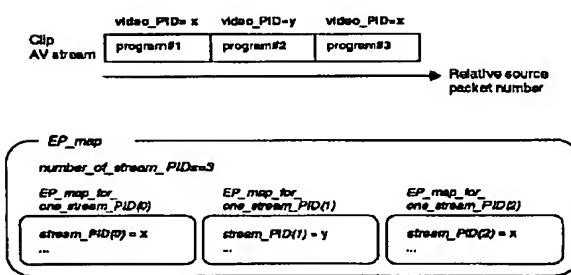
【図63】



【図80】

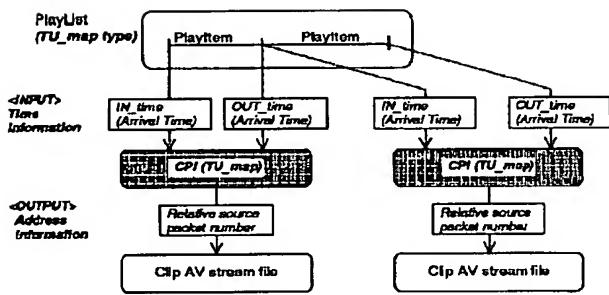


【図69】



Syntax	No. of bits	Mnemonics
EP_map0{		
reserved	12	bsb1
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID(k)	16	bsb1
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bsb1
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (i=0;i<Y;i++){		
padding_word	16	bsb1
}		
}		
}		

【図64】



【図88】

Thumbnail_picture_format	Meaning
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0x1f	reserved

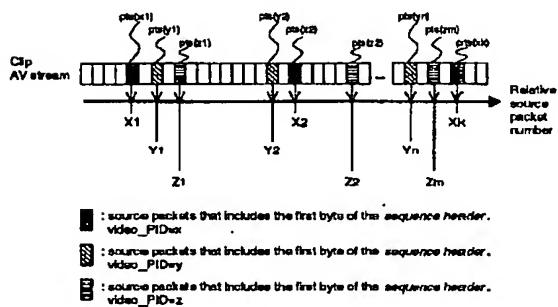
thumbnail_picture_format

【図95】

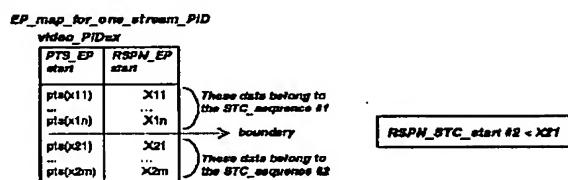
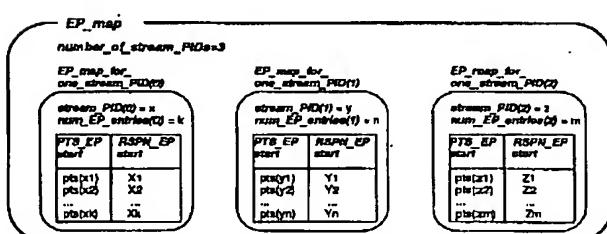
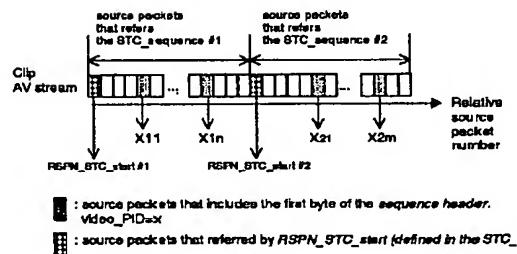
copy_permission_indicator	meaning
00	copy free
01	no more copy
10	copy once
11	copy prohibited

copy permission indicator table

【図67】



【図68】



ビデオのEP_mapの例

【図72】

Syntax	No. of bits	of	Mnemonics
EP_map_for_one_stream_PID(N)			
for (i=0; i< N; i++) {			
PTS_EP_start	32	uimsbf	
RSPN_EP_start	32	uimsbf	
}			
}			

Syntax	No. of bits	Mnemonics
ClipMark()		
version_number	8*4	bsbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i< number_of_Clip_marks; i++) {		
reserved	8	bsbf
mark_type	8	bsbf
reserved_for_MarkerID	16	bsbf
mark_entry()		
representative_picture_entry()		
ref_thumbnail_index	16	uimsbf
}		
)		

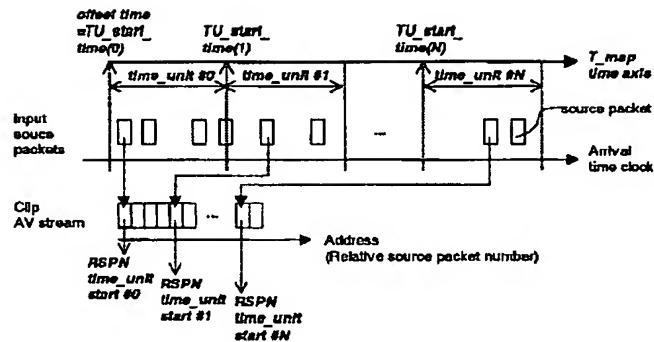
EP_map_for_one_stream_PIDのシンタクス

【図71】

EP_type	Meaning
0	video
1	audio
2 - 15	reserved

EP_type Values

【図73】



【図81】

Syntax	No. of bits	Mnemonics
mark_entry() / representative_picture_entry()		
mark_time_stamp	32	ulmsbf
STC_sequence_id	8	ulmsbf
reserved	24	bsbf
}		

【図74】

Syntax	No. of bits	Mnemonics
TU_map()		
offset_time	32	bsbf
time_unit_size	32	ulmsbf
number_of_time_unit_entries	32	ulmsbf
for (k=0; k<number_of_time_unit_entries; k++)		
RSPN_time_unit_start	32	ulmsbf
}		

TU_map のシンタクス

【図75】

【図79】

Syntax	No. of bits	Mnemonics	Mark_type	Meaning	Comments
ClipMark()			0x00 - 0x8F	reserved	Reserved for PlaylistMark()
version_number	8*4	bsbf	0x90	Event-start mark	番組の開始ポイントを示すマーク点
length	32	ulmsbf	0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点
number_of_Clip_marks	16	ulmsbf	0x92	Scene-start mark	シーン開始ポイントを示すマーク点
for (i=0; i<number_of_Clip_marks; i++)			0x93	Scene-end mark	シーン終了ポイントを示すマーク点
reserved	8	bsbf	0x94	CM-start mark	CM開始ポイントを示すマーク点
mark_type	8	bsbf	0x95	CM-end mark	CM終了ポイントを示すマーク点
mark_time_stamp	32	ulmsbf	0x96 - 0xBF	DVR フォーマットが、 ClipMark を将来、拡張する時のため予約されている	
STC_sequence_id	8	ulmsbf	0xC0 - 0xFF	メーカー独自のアプリケーションで利用するマークに割り当て可能	
reserved	24	bsbf			
character_set	8	bsbf			
name_length	8	ulmsbf			
mark_name	8*256	bsbf			
ref_thumbnail_index	16	ulmsbf			
}					

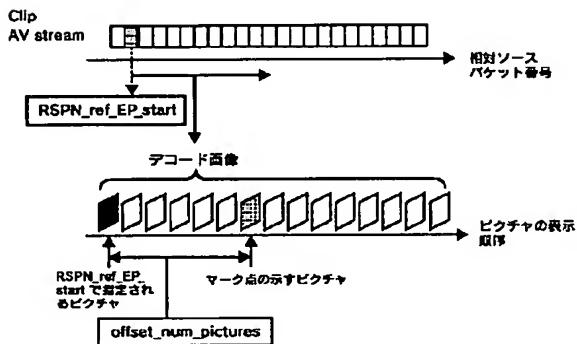
ClipMark のシンタクス

【図77】

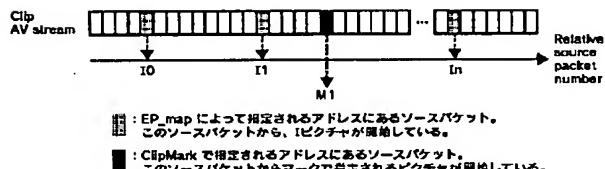
CPI_type in the CPI0	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めで表さなければならない。mark_time_stamp は、次に示す式により計算される。 $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

mark_type_stamp

【図83】



【図85】



【図86】

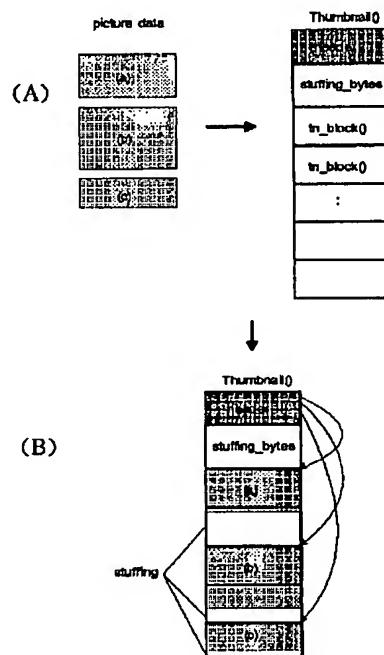
Syntax	No. of bits	Mnemonics
mark_entry() / representative_picture_entry()		
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	92	uimsbf

【図82】

【図84】

Syntax	No. of bits	Mnemonics
mark_entry() / representative_picture_entry()		
RSPN_mark_point	32	uimsbf

【図89】



Syntax	No. of bits	Mnemonics
menu.thmb / mark.thmb {		
reserved	256	bslbf
Thumbnail0		
for(j=0; j<N1; i++)		
padding_word	16	bslbf
}		

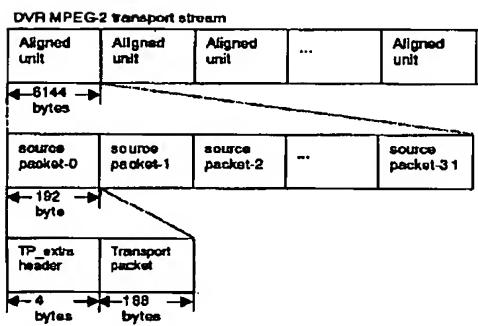
menu.thmb と mark.thmb のシンタクス

【図87】

Syntax	Bits	Mnemonics
Thumbnail() {		
version number	8*4	char
length	32	ulmsbf
if (length != 0) {		
tn_blocks_start_address	32	belbf
number_of_thumbnails	16	ulmsbf
tn_block_size	16	ulmsbf
number_of_tn_blocks	16	ulmsbf
reserved	16	belbf
for(i = 0; i < number_of_thumbnails; i++) {		
thumbnail_index	16	ulmsbf
thumbnail_picture_format	8	bslbf
reserved	8	belbf
picture_data_size	32	ulmsbf
start_tn_block_number	16	ulmsbf
x_picture_length	16	ulmsbf
y_picture_length	16	ulmsbf
reserved	16	ulmsbf
}		
stuffing_bytes	8*2*L1	belbf
for(k = 0; k < number_of_tn_blocks; k++) {		
tn_block	tn_block_size* 1024*8	
}		
}		
}		

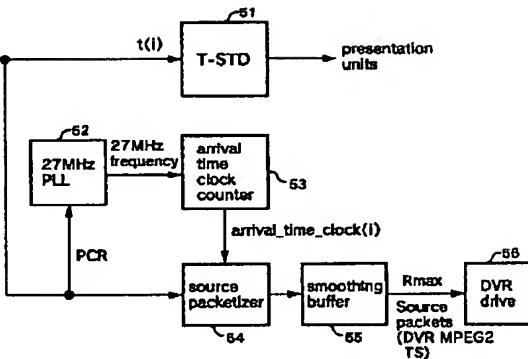
Thumbnail のシンタクス

【図90】



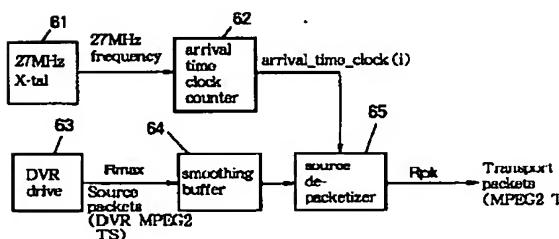
DVR MPEG-2 トランスポートストリームの構造

【図91】



DVR MPEG-2 トランスポートストリームのレコーダモデル

【図92】



DVR MPEG-2 トランスポートストリームのプレーヤモデル

【図93】

Syntax	No. of bits	Mnemonics
source_packet() {		
TP_extra_header()		
transport_packet()		

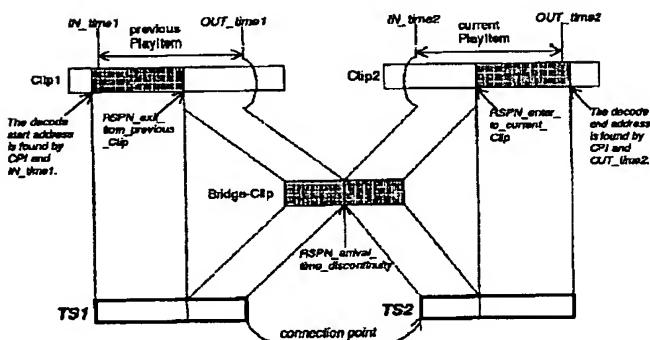
source_packet

【図94】

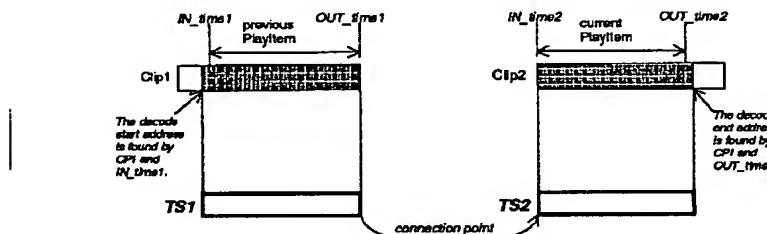
Syntax	No. of bits	Mnemonics
TP_extra_header()		
copy_permission_indicator	2	ulmsbf
arrival_time_stamp	30	ulmsbf
}		

TP_extra_header

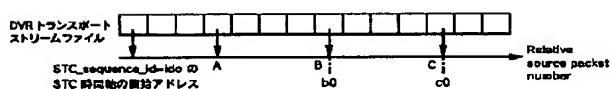
【図96】



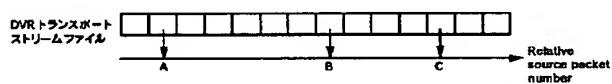
【図97】



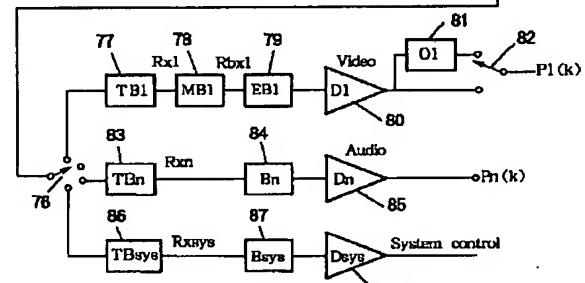
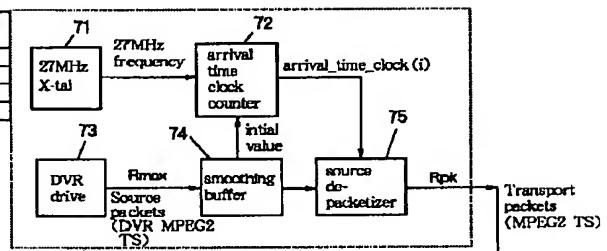
【図107】



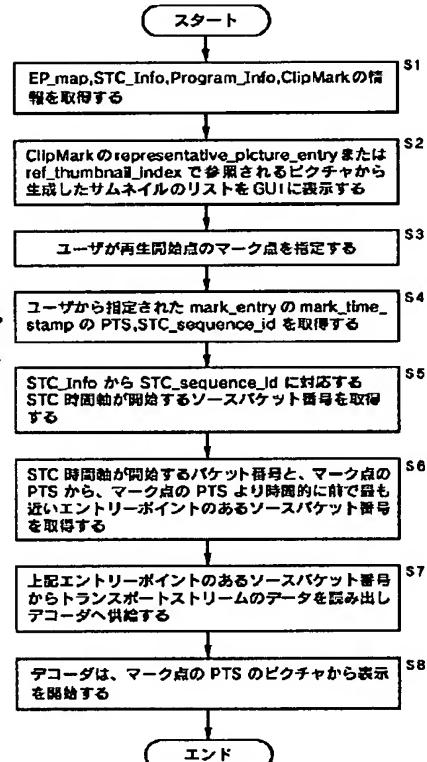
【図113】



【図104】

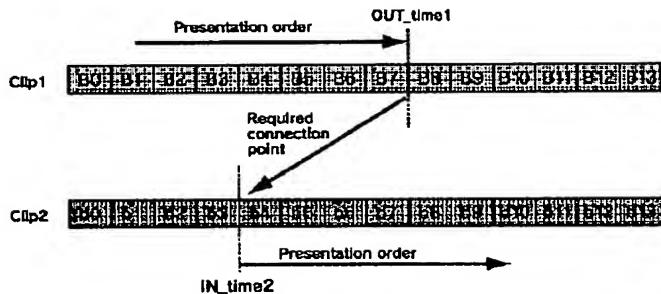


【図106】



エンド

【図98】



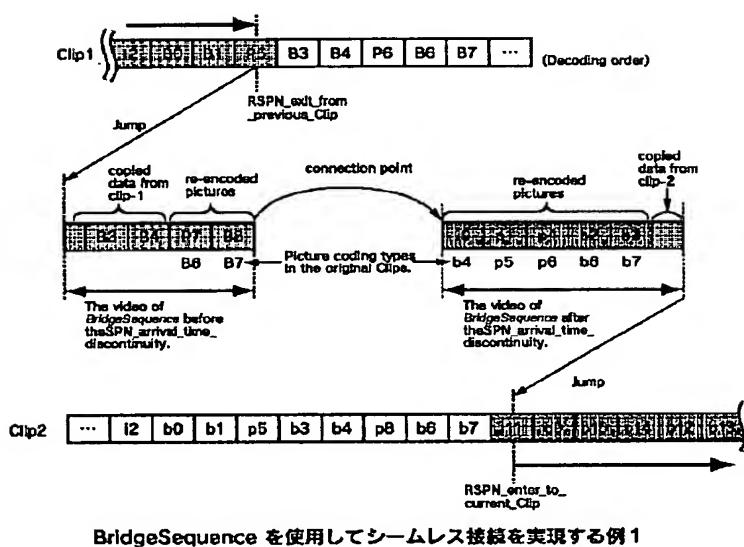
ClipMark

Mark_type	mark_entry		representative_picture_entry	
	Mark_Time_stamp	STC_sequence_id	Mark_Time_stamp	STC_sequence_id
...
0x92(scene start)	PTS(a1)	id0	PTS(a2)	id0
0x94(Chstart)	PTS(b0)	id0	PTS(b0)	id0
0x95(Chend)	PTS(c0)	...	PTS(c0)	...
...

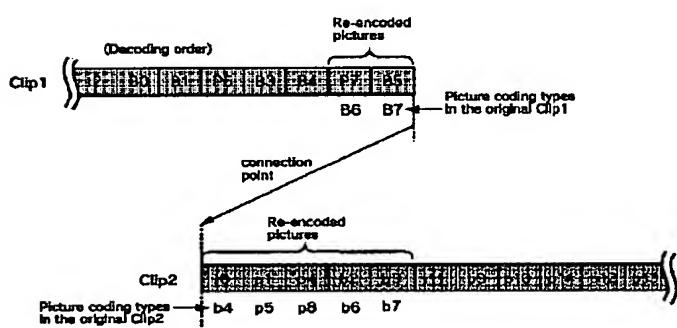
【図109】

ピクチャの表示順序で示すシームレス接続の例

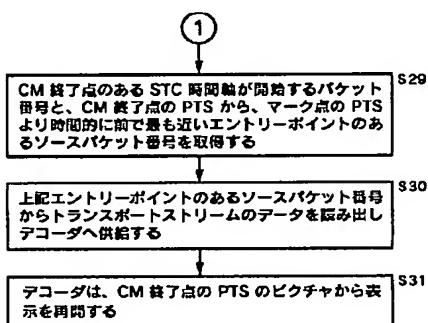
【図99】



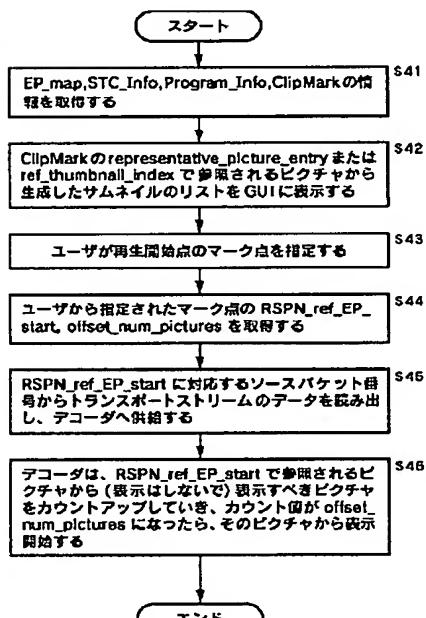
【図100】



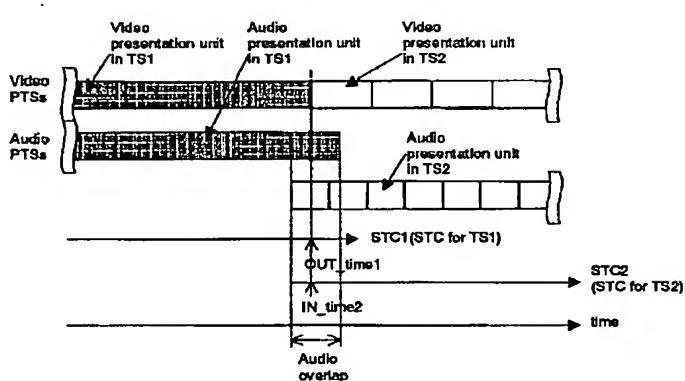
【図111】



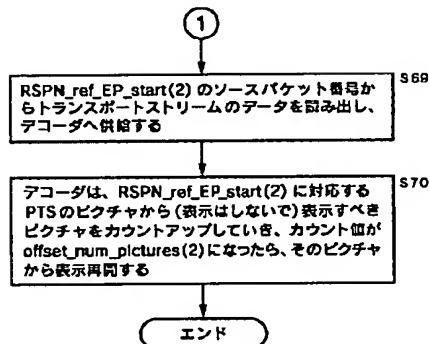
【図112】



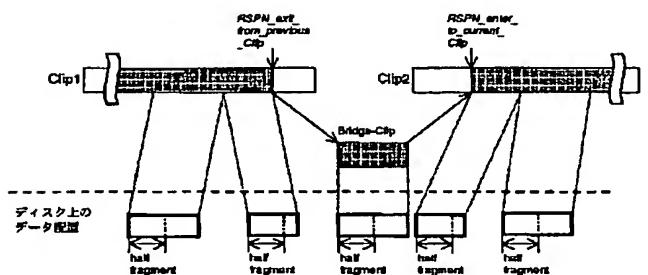
【図101】



【図117】

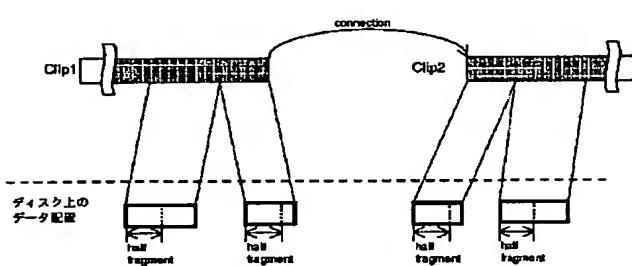


【図102】



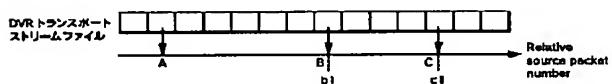
BridgeSequence を使用してシームレス接続をする場合の、データアロケーションの例

【図103】

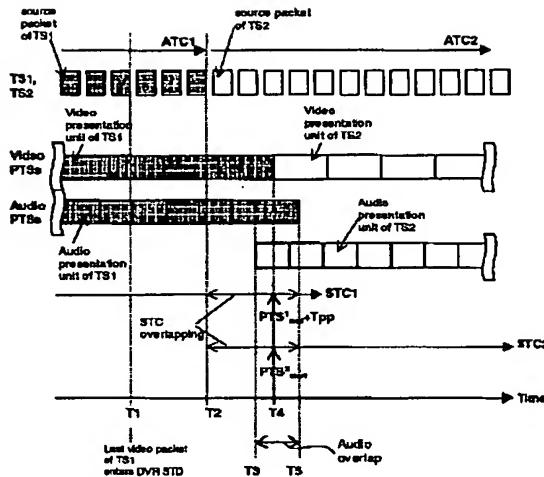


BridgeSequence を使用しないでシームレス接続をする場合の、データアロケーションの例

【図119】

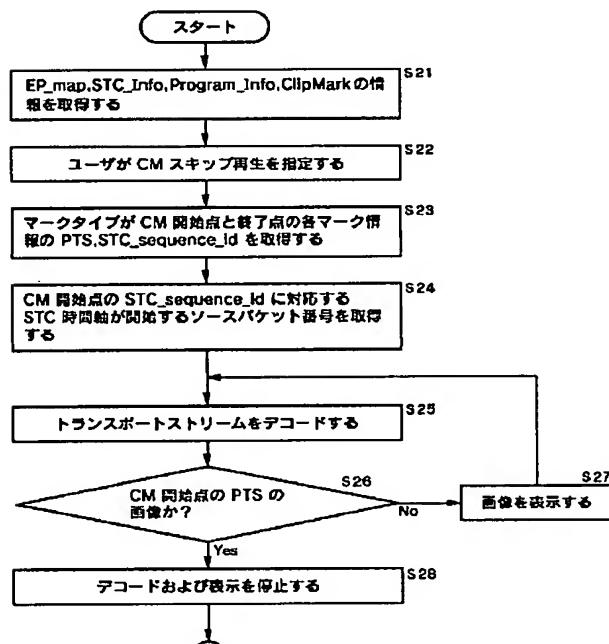


【図105】



ある AV ストリーム(TS1)からそれにシームレスに接続された次の AV ストリーム(TS2)
へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャート

【図110】



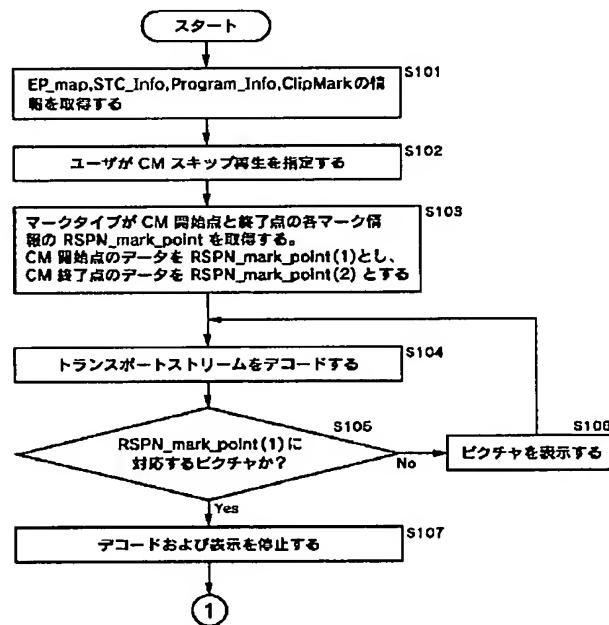
【図121】

mark_type	mark_entry	representative_picture_entry
	RSPN_mark_point	RSPN_mark_point
...
0x92(scene start)	a1	a2
0x94(CM start)	b1	b1
0x95(CM end)	c1	c1
...

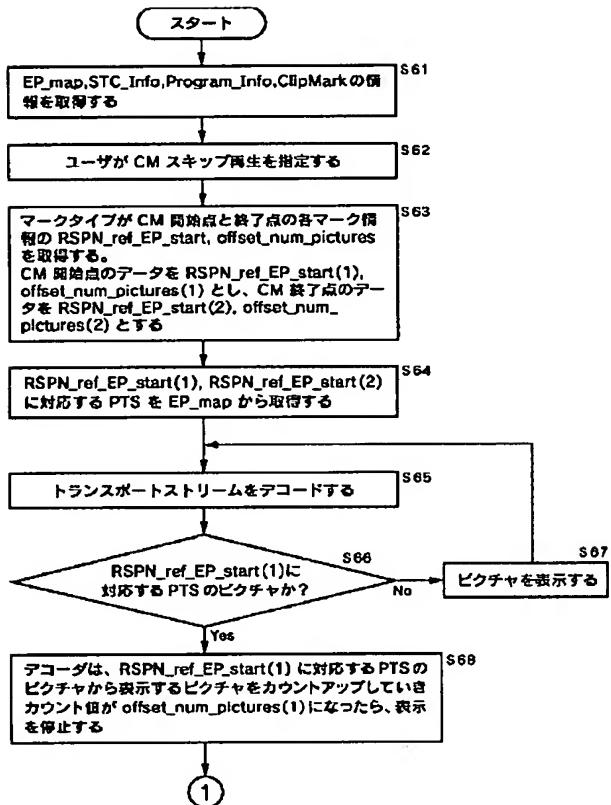
【図115】

mark_type	mark_entry		representative_picture_entry	
	RSPN_ref_EP_start	offset_num_pictures	RSPN_ref_EP_start	offset_num_pictures
...
0x92(scene start)	A	M1	A	M2
0x94(CM start)	B	N1	B	N1
0x95(CM end)	C	N2	C	N2
...

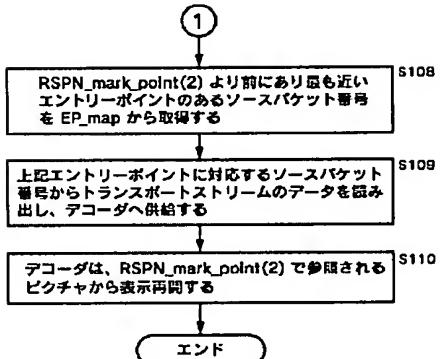
【図122】



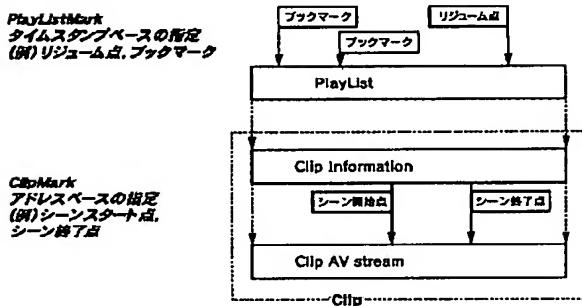
【图116】



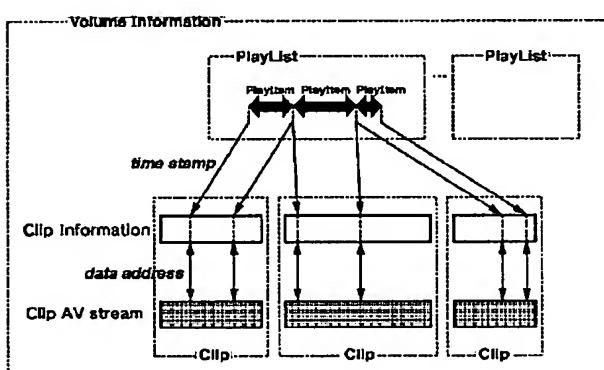
【図123】



【125】



【図124】



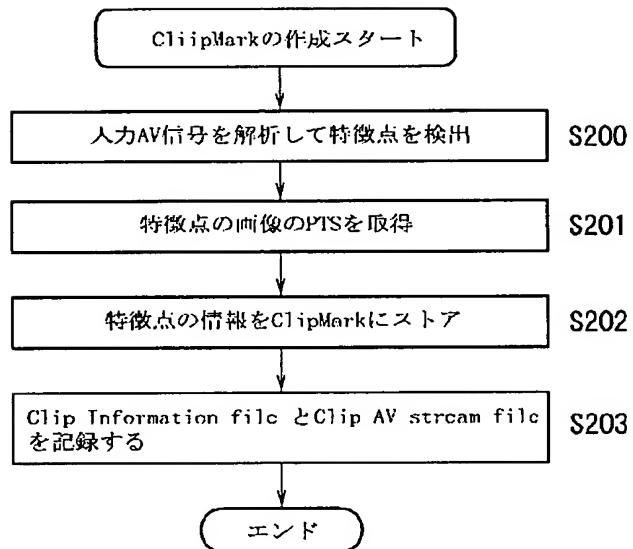
【图 126】

Syntax	No. of bits	Mnemonics
ClipMark0 {		
version_number	8*4	bsbf
length	32	ulmsbf
number_of_Clip_marks	16	ulmsbf
for (i=0;i<number_of_Clip_marks;i++){		
reserved	8	bsbf
mark_type	8	bsbf
RSPN_mark	32	ulmsbf
reserved	32	bsbf
ref_thumbnail_index	16	ulmsbf
}		

[図127]

Syntax		No. of bits	Mnemonics
ClipMark0 {			
version_number		8*4	bslbf
length		32	umsbsf
number_of_Clip_marks		16	umsbsf
for (i=0;<number_of_Clip_marks; i++) {			
reserved		8	bslbf
mark_type		8	bslbf
RSPN_ref_EP_start		32	umsbsf
offset_num_pictures		32	umsbsf
ref_thumbnail_index		16	umsbsf
}			

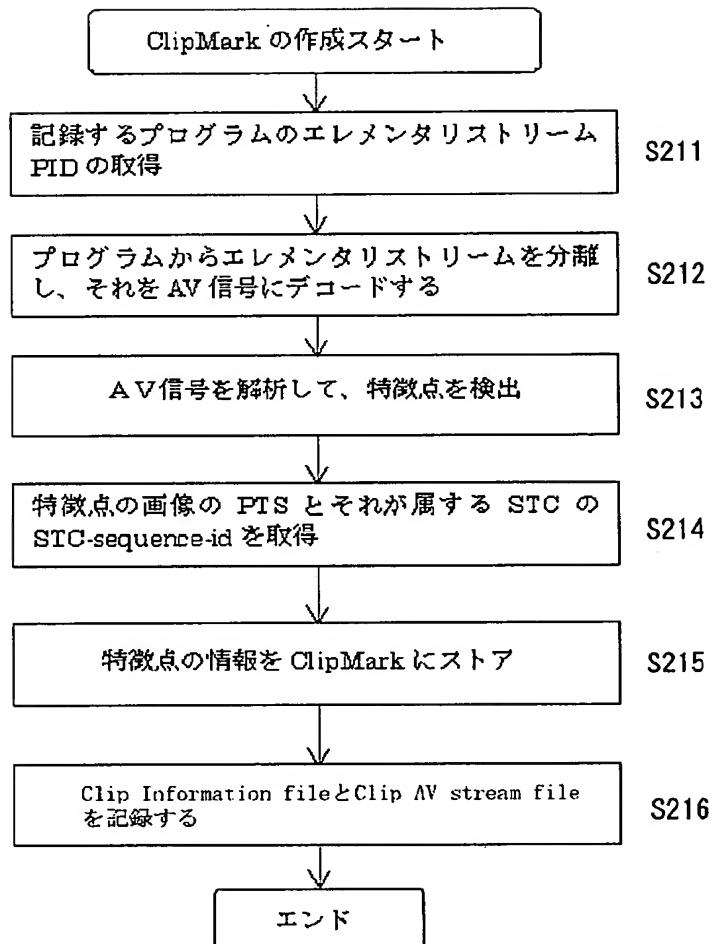
【図128】



【図130】



【図129】

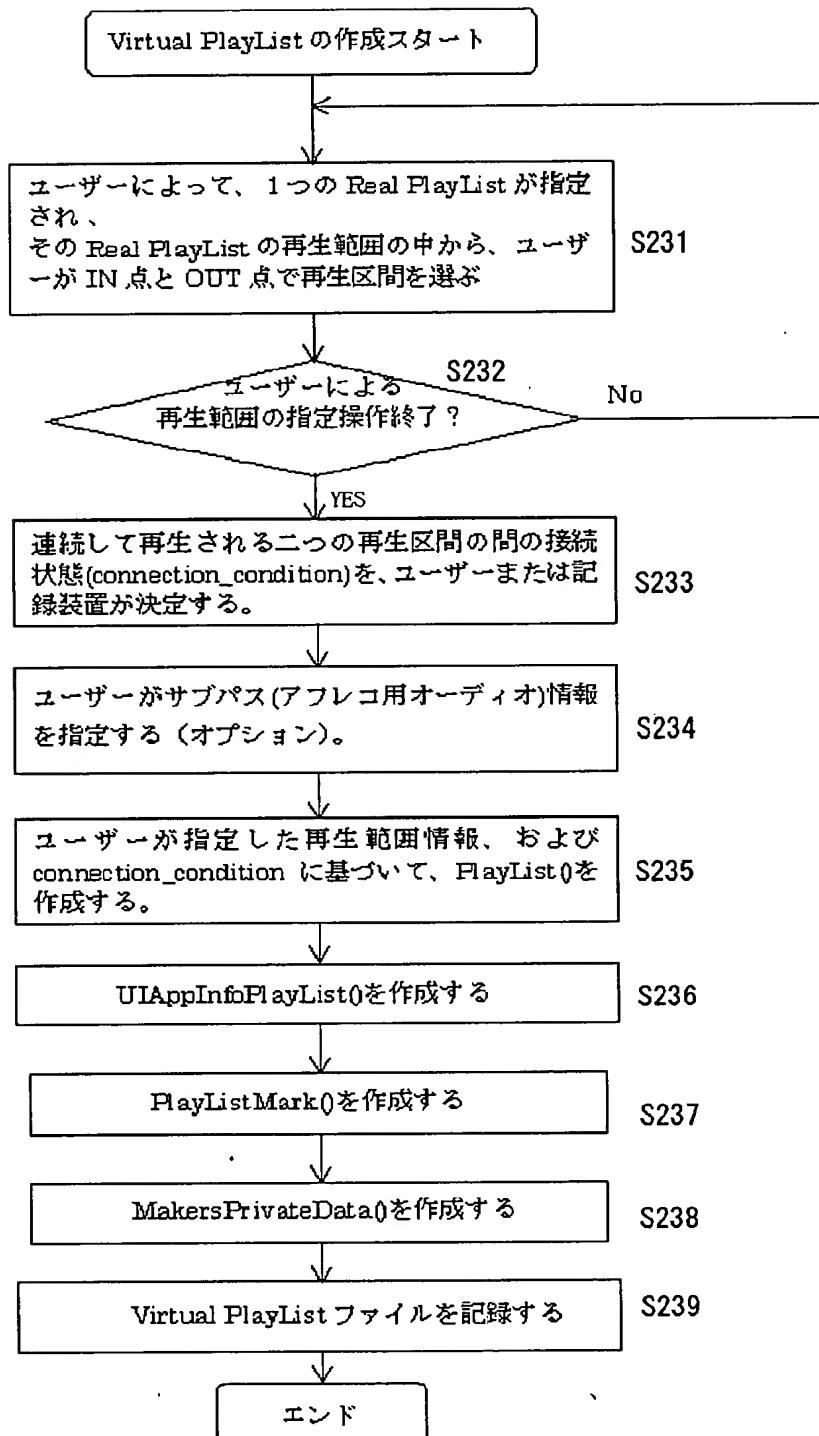


【図135】

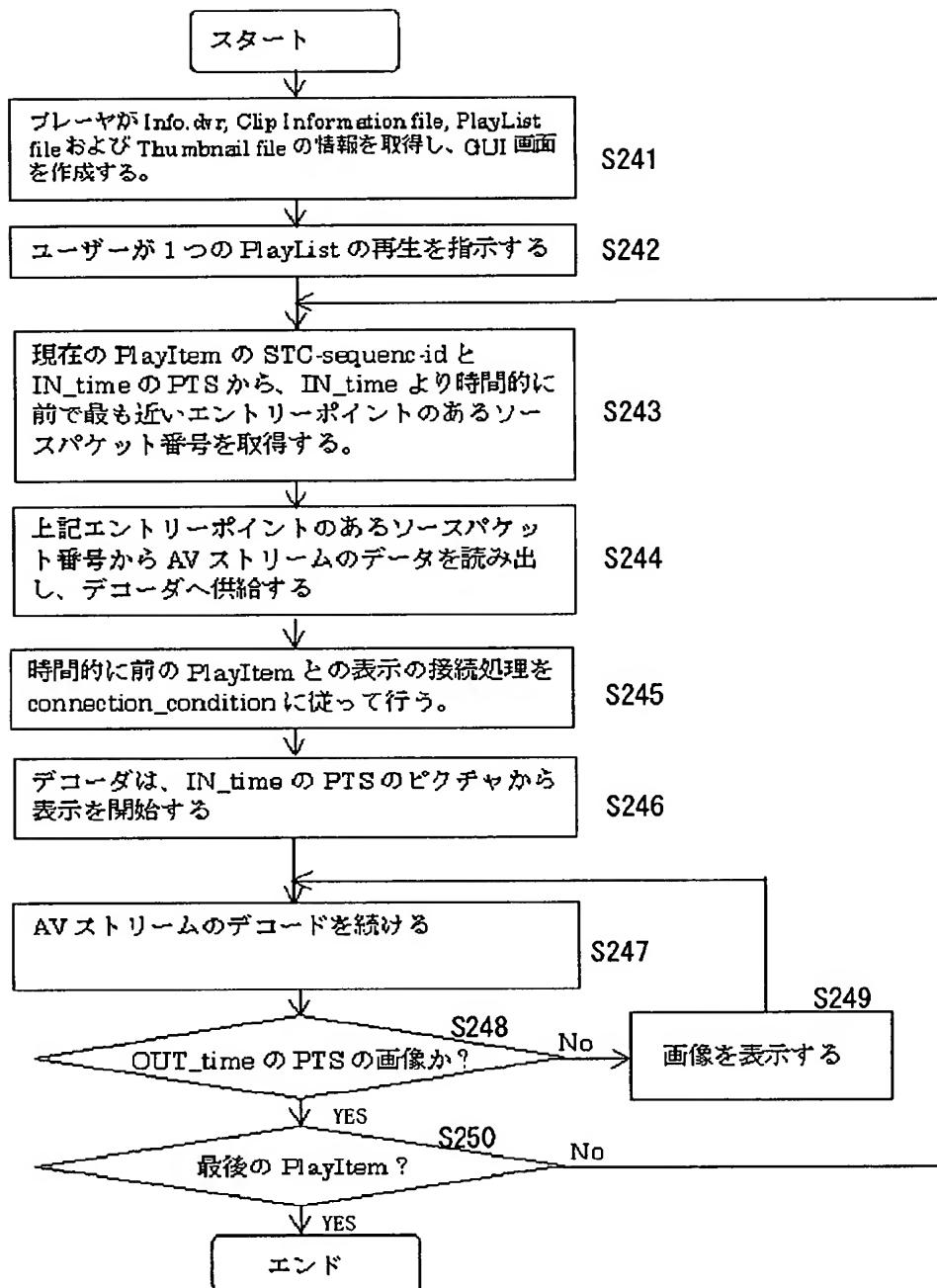
Syntax	No. of bits	Mnemonic
PlayListMark()		
length	32	ulmsbf
number_of_PlayList_marks	16	ulmsbf
for(i=0; i < number_of_PlayList_marks; i++) {		
mark_invalid_flag	1	ulmsbf
mark_type	7	ulmsbf
mark_name_length	8	ulmsbf
ref_to_PlayList_id	16	ulmsbf
mark_time_stamp	32	ulmsbf
entry_ES_PID	16	ulmsbf
ref_to_thumbnail_index	16	ulmsbf
mark_name	8*32	bsibf
}		
}		

PlayListMark() のシンタクスの別例

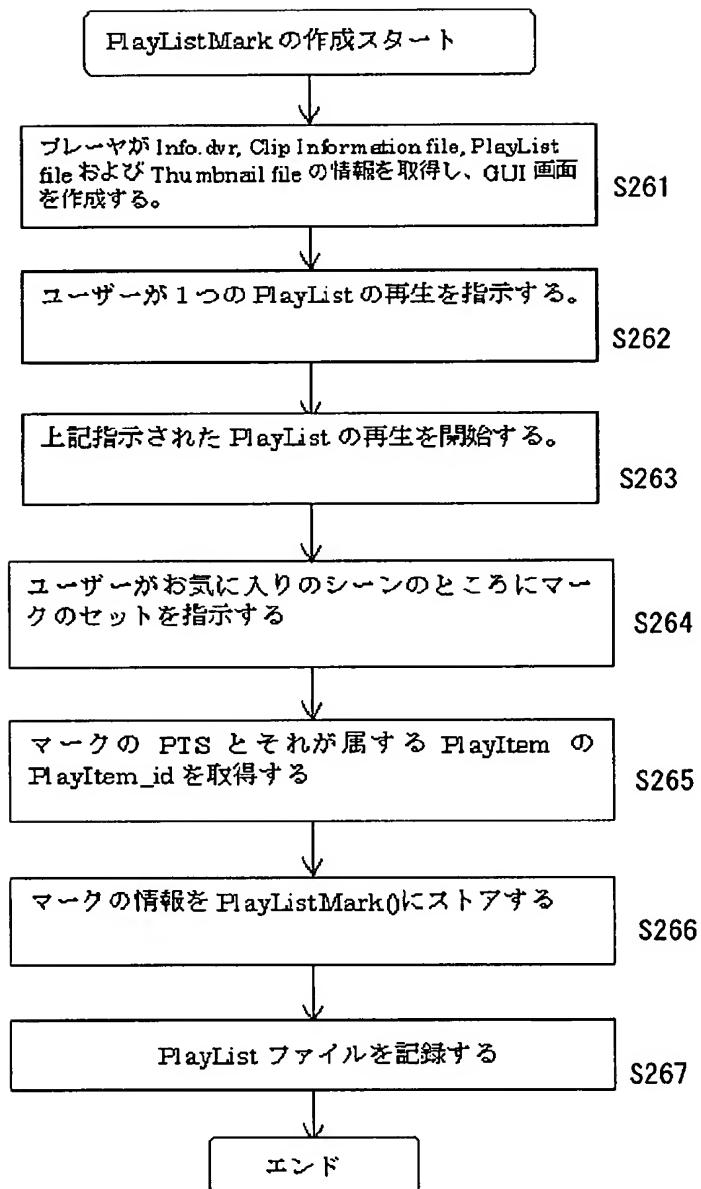
【図131】



【図132】



【図133】

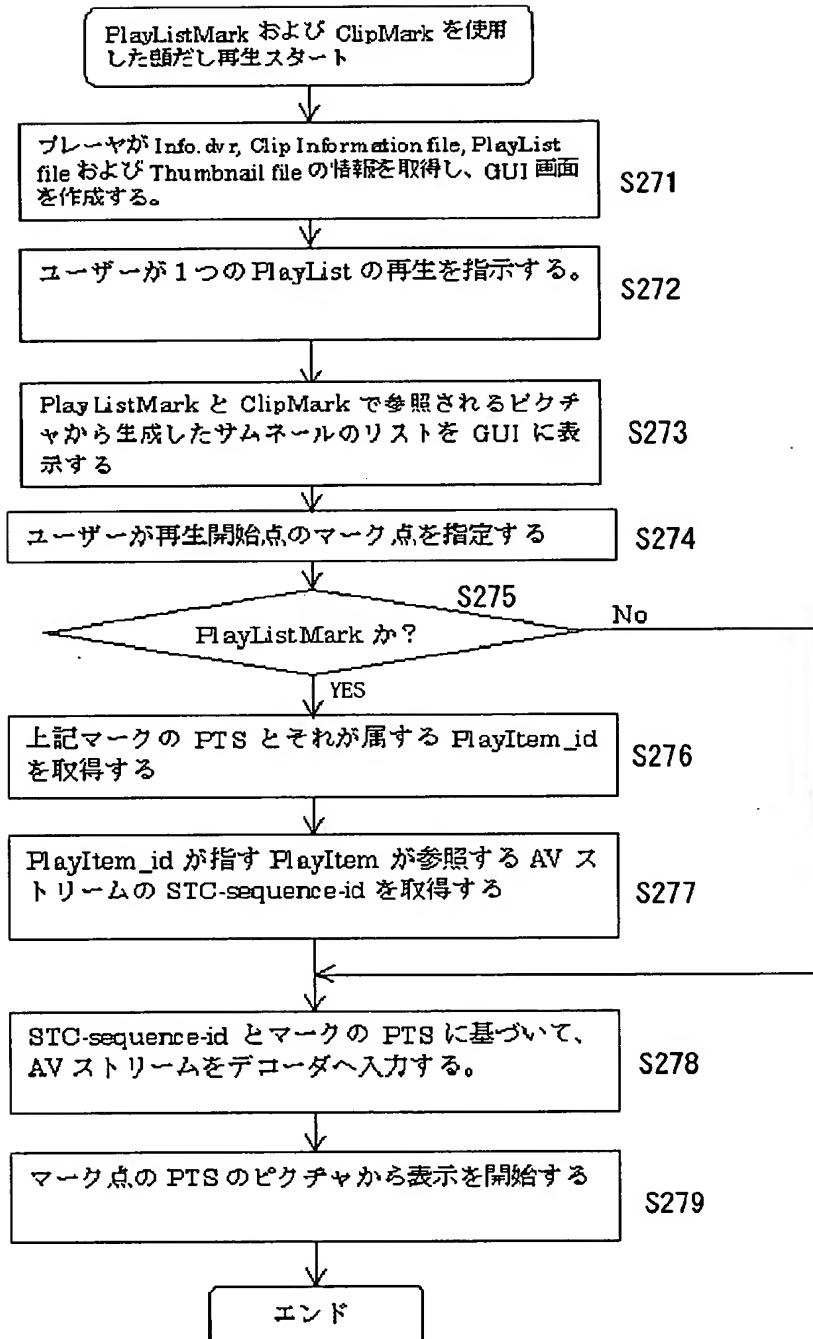


【図138】

Mark_type	Meaning	Note
0x00 - 0x3F	reserved for future use	Reserved for PlayListMark
0x40	Scene-start-mark	シーンの開始ポイントを示すマーク点。
0x41 - 0x5F	Reserved for common ClipMark	
0x60 - 0x7F	Marker defined ClipMark	marker_ID によって示されるメーカーが自由に意味を定義できる。

ClipMark() の mark_type の意味を説明するテーブル

【図134】



【図136】

value	Meaning	Note
0x00	Resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	Book-mark	PlayList の再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークを使う。このマークは、PlayListMark()に複数あっても良い。
0x02	Chapter-mark	ユーザーは、PlayList の中で1つのチャプターがこのマークから開始することを意図している。ユーザがセットすることができる。このマークは、PlayListMark()に複数あっても良い。
0x03	Skip-start-mark	PlayListMark の中に1つの Skip-start-mark がセットされる場合、その Skip-start-mark のエントリーの直後に1つの Skip-end-mark がセットされなければならない。
0x04	Skip-end-mark	Skip-start-mark のタイムスタンプから Skip-end-mark のタイムスタンプまで、ユーザーは、PlayList の再生をスキップすることを意図している。 Skip-start-mark と Skip-end-mark は、同じ ref_to_PlayItem_id を持つ。また、Skip-start-mark と Skip-end-mark は、もし entry_ES_PID が 0xFFFF でないならば、同じ entry_ES_PID の値を持つ。 ユーザがセットすることができるマークであり、このマークは、PlayListMark()に複数あっても良い。
0x05 - 0x3F	Reserved for future use	Reserved for PlayListMark
0x40 - 0x7F	Reserved for ClipMark	

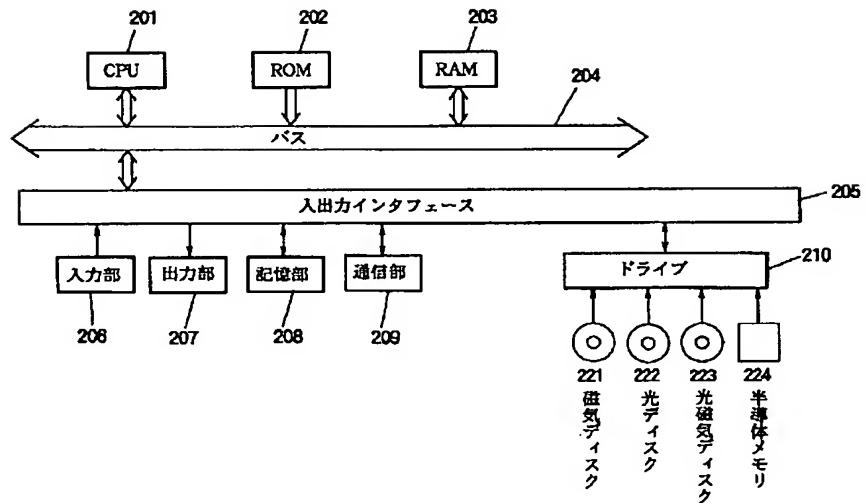
PlayListMark()のmark_typeの意味を 説明するテーブル

【図137】

Syntax	No. of bits	Mnemonic
ClipMark()		
length	32	uimsbf
marker_ID	16	uimsbf
number_of_Clip_marks	16	ulmabf
for(i=0; i < number_of_Clip_marks; i++) {		
mark_invalid_flag	1	uimsbf
mark_type	7	uimsbf
ref_to_STC_id	8	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_PID	16	uimsbf
ref_to_thumbnail_index	16	uimsbf
representative_picture_time_stamp	32	uimsbf
}		

ClipMark()の別例

【図139】



フロントページの続き

Fターム(参考) 5C052 AA02 AB03 AB04 AC08 CC06
 CC11 DD04
 5C053 FA14 FA23 GB05 GB38 HA29
 JA16 JA22 JA24 LA04 LA05
 LA11